

## CBSA Restful API Guide.

We supply two interfaces which drive the restful API for CBSA under the covers; the Payment interface and the Customer Service interface. These interfaces have different usage scenarios, aimed at different audiences.

The Restful APIs for both can be driven directly from a URL, or can be driven from the provided Payment and Customer Services Interfaces. In each case the URL goes via a zOS Connect EE server (represented below as ***your-host-name*** and ***your-port-number***)

### The PAYMENT interface:

Think of the Payment interface as being similar to a debit or credit card machine that you'd find in a business. Instead of swiping a card, the business has access to take payments or make refunds, via the interface, for goods and services, directly from their customer's bank account. For further details about the Payment interface please refer to the Payment Interface User Guide.

### The associated restful Payment details:

The Payment interface utilises a new z/OS Connect EE restful API, which requires the use of a zOS Connect EE server. The restful API can also be driven independently of the Payment interface. It can, therefore, be integrated into other applications if desired.

#### The Payment Service (Pay) details are:

```
"ServiceName": "Pay",  
"ServiceDescription": "CBSA Pay Service",  
"ServiceProvider": "CICS-1.0",  
"ServiceURL": "http://your-host-name:your-port-number  
/zosConnect/services/Pay"
```

#### The Payment Service APIs:

There is only one API available for the Payment Service, and that is the ***makepayment*** API. The details are as follows:

```
"name": "makepayment",  
"version": "1.0.0",  
"description": "Make a payment on the Pay Service",  
"adminUrl": http://your-host-name:your-port-number/zosConnect/apis/makepayment
```

**The URL to drive restful Payments is:**

**`http://your-host-name:your-port-number/makepayment/dbcr`**

The Payment URL issues an HTTP PUT and requires a mandatory JSON request body (containing details of the account number (the payment is to be taken from), the company taking the payment and the payment amount).

**A valid JSON *request* body example:**

```
{
  "PAYDBCR": {
    "COMM_ACCNO": "00000876",
    "COMM_AMT": -10.99,
    "COMM_SORTC": 000000,
    "COMM_AV_BAL": 0,
    "COMM_ACT_BAL": 0,
    "COMM_ORIGIN": {
      "COMM_APPLID": "NETFLIX ",
      "COMM_USERID": "LTD      ",
      "COMM_FACILITY_NAME": "      ",
      "COMM_NETWRK_ID": "      ",
      "COMM_FACILTYPE": 496,
      "FILL_0": "      "
    },
    "COMM_SUCCESS": " ",
    "COMM_FAIL_CODE": " "
  }
}
```

This example JSON request body is used by the company NETFLIX LTD to request a Payment of -10.99 (a payment is represented by a negative COMM\_AMT value) from account number 00000876 at the bank.

**NOTE** The request body is mandatory and all fields should be provided as detailed above (it is important that the COMM\_FACILTYPE is set to 496, to internally differentiate the PAYMENT traffic from BMS traffic). The company name may be up to 16 bytes in length and can be spread across COMM\_APPLID and COMM\_USERID (as shown above).

**The JSON *response* body returned is:**

```
{ "PAYDBCR": {
  "COMM_ORIGIN": {
    "COMM_FACILTYPE": 496,
    "COMM_NETWRK_ID": "",
    "COMM_APPLID": "NETFLIX",
    "COMM_FACILITY_NAME": "",
    "COMM_USERID": "LTD",
    "FILL_0": ""
  },
  "COMM_AV_BAL": 700382.9,
  "COMM_FAIL_CODE": "0",
```

```

"COMM_SORTC": 987654,
"COMM_AMT": -10.99,
"COMM_ACCNO": "00000876",
"COMM_ACT_BAL": 700382.9,
"COMM_SUCCESS": "Y"
}}

```

In this instance the request was successful, denoted by the COMM\_SUCCESS attribute containing 'Y'. On the response body, integer fields do not have leading zeros. We have intentionally chosen to utilise the full copybook on the JSON request and response, illustrating the full range of data which may be flowed. Should some attributes not be required, it is simple enough to change the mappings in zOS Connect EE.

### Examples of the JSON *response* body returned if the payment/credit cannot be made:

There are occasions when a payment/credit may not be successful, for example:

1. *Account not found*: In this example account number 99999876 does not exist. Here is the JSON response body returned under this circumstance:

```

{"PAYDBCR": {
  "COMM_ORIGIN": {
    "COMM_FACILTYPE": 496,
    "COMM_NETWRK_ID": "",
    "COMM_APPLID": "NETFLIX",
    "COMM_FACILITY_NAME": "",
    "COMM_USERID": "LTD",
    "FILL_0": ""
  },
  "COMM_AV_BAL": 0,
  "COMM_FAIL_CODE": "1",
  "COMM_SORTC": 987654,
  "COMM_AMT": -10.99,
  "COMM_ACCNO": "99999876",
  "COMM_ACT_BAL": 0,
  "COMM_SUCCESS": "N"
}}

```

The failure is denoted by the COMM\_SUCCESS being set to 'N' and a COMM\_FAIL-CODE of '1' indicates that the account number could not be found.

2. *A problem with Db2*:

The JSON response body returned:

```

{"PAYDBCR": {
  "COMM_ORIGIN": {
    "COMM_FACILTYPE": 496,
    "COMM_NETWRK_ID": "",
    "COMM_APPLID": "NETFLIX",
    "COMM_FACILITY_NAME": "",
    "COMM_USERID": "LTD",
    "FILL_0": ""
  }
}

```

```

    },
    "COMM_AV_BAL": 0,
    "COMM_FAIL_CODE": "2",
    "COMM_SORTC": 987654,
    "COMM_AMT": -10.99,
    "COMM_ACCNO": "00000777",
    "COMM_ACT_BAL": 0,
    "COMM_SUCCESS": "N"
  }}

```

The failure is denoted by the COMM\_SUCCESS being set to 'N' and a COMM\_FAIL-CODE of '2' indicates that there was a problem in Db2 and that the request could not be successfully completed.

### 3. Requesting a payment from a LOAN or MORTGAGE account:

The JSON response body returned:

```

{"PAYDBCR": {
  "COMM_ORIGIN": {
    "COMM_FACILTYPE": 496,
    "COMM_NETWRK_ID": "",
    "COMM_APPLID": "NETFLIX",
    "COMM_FACILITY_NAME": "",
    "COMM_USERID": "LTD",
    "FILL_0": ""
  },
  "COMM_AV_BAL": 0,
  "COMM_FAIL_CODE": "4",
  "COMM_SORTC": 987654,
  "COMM_AMT": -10.99,
  "COMM_ACCNO": "00000005",
  "COMM_ACT_BAL": 0,
  "COMM_SUCCESS": "N"
}}

```

The failure is denoted by the COMM\_SUCCESS being set to 'N' and a COMM\_FAIL-CODE of '4' indicates that the user is trying to make a payment request against either a LOAN or MORTGAGE account, to do so does not make sense and is therefore prohibited.

### 4. Requesting a payment from an account with insufficient funds available:

The JSON response body returned:

```

{"PAYDBCR": {
  "COMM_ORIGIN": {
    "COMM_FACILTYPE": 496,
    "COMM_NETWRK_ID": "",
    "COMM_APPLID": "NETFLIX",
    "COMM_FACILITY_NAME": "",
    "COMM_USERID": "LTD",

```

```

        "FILL_0": ""
    },
    "COMM_AV_BAL": 0,
    "COMM_FAIL_CODE": "3",
    "COMM_SORTC": 987654,
    "COMM_AMT": -10.99,
    "COMM_ACCNO": "00001490",
    "COMM_ACT_BAL": 0,
    "COMM_SUCCESS": "N"
}}

```

The failure is denoted by the COMM\_SUCCESS being set to 'N' and a COMM\_FAIL-CODE of '3' indicates that there are insufficient funds available in the account and the requested payment is therefore not granted.

#### Other information:

The same interface can also be used by NETFLIX LTD to issue a credit back to a customer. To credit an amount simply drop the minus sign from the COMM\_AMT.

#### What is being driven at the backend by this API?:

Backend program **DBCRFUN** is being driven by this API call utilising copybook **PAYDBCR**.

#### Note of caution:

The old saying “garbage in / garbage out” applies. The RestFul Payment API should be used in accordance with the instructions as detailed above. Where possible it is recommended that the Payment API is integrated into an application, where validation checking of the JSON request and response bodies can be made. We perform such checking in the supplied Payment Interface. Scenarios outside of this, may need to perform additional validation checks to avoid data corruption.

#### Swagger details:

```

{
  "swagger": "2.0",
  "info": {
    "description": "Make a payment on the Pay Service",
    "version": "1.0.0",
    "title": "makepayment"
  },
  "basePath": "/makepayment",
  "schemes": [

```

```

    "https",
    "http"
  ],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {"/dbcr": {"put": {
    "tags": ["makepayment"],
    "operationId": "putPay",
    "parameters": [
      {
        "name": "Authorization",
        "in": "header",
        "required": false,
        "type": "string"
      },
      {
        "in": "body",
        "name": "putPay_request",
        "description": "request body",
        "required": true,
        "schema": {"$ref": "#/definitions/putPay_request"}
      }
    ],
    "responses": {"200": {
      "description": "OK",
      "schema": {"$ref": "#/definitions/putPay_response_200"}
    }}
  }},
  "definitions": {
    "putPay_request": {
      "type": "object",
      "properties": {"PAYDBCR": {
        "type": "object",
        "properties": {
          "COMM_ACCNO": {
            "type": "string",
            "maxLength": 8
          }
        }
      }
    }
  }
}

```

```

},
"COMM_AMT": {
  "type": "number",
  "format": "decimal",
  "minimum": -9.999999999999E9,
  "maximum": 9.999999999999E9
},
"COMM_SORTC": {
  "type": "integer",
  "minimum": 0,
  "maximum": 999999
},
"COMM_AV_BAL": {
  "type": "number",
  "format": "decimal",
  "minimum": -9.999999999999E9,
  "maximum": 9.999999999999E9
},
"COMM_ACT_BAL": {
  "type": "number",
  "format": "decimal",
  "minimum": -9.999999999999E9,
  "maximum": 9.999999999999E9
},
"COMM_ORIGIN": {
  "type": "object",
  "properties": {
    "COMM_APPLID": {
      "type": "string",
      "maxLength": 8
    },
    "COMM_USERID": {
      "type": "string",
      "maxLength": 8
    },
    "COMM_FACILITY_NAME": {
      "type": "string",

```

```

        "maxLength": 8
    },
    "COMM_NETWRK_ID": {
        "type": "string",
        "maxLength": 8
    },
    "COMM_FACILTYPE": {
        "type": "integer",
        "minimum": -999999999,
        "maximum": 999999999
    },
    "FILL_0": {
        "type": "string",
        "maxLength": 4
    }
},
"COMM_SUCCESS": {
    "type": "string",
    "maxLength": 1
},
"COMM_FAIL_CODE": {
    "type": "string",
    "maxLength": 1
}
}}
},
"putPay_response_200": {
    "type": "object",
    "properties": {"PAYDBCR": {
        "type": "object",
        "properties": {
            "COMM_ACCNO": {
                "type": "string",
                "maxLength": 8
            },

```



```

"COMM_AMT": {
  "type": "number",
  "format": "decimal",
  "minimum": -9.999999999999E9,
  "maximum": 9.999999999999E9
},
"COMM_SORTC": {
  "type": "integer",
  "minimum": 0,
  "maximum": 999999
},
"COMM_AV_BAL": {
  "type": "number",
  "format": "decimal",
  "minimum": -9.999999999999E9,
  "maximum": 9.999999999999E9
},
"COMM_ACT_BAL": {
  "type": "number",
  "format": "decimal",
  "minimum": -9.999999999999E9,
  "maximum": 9.999999999999E9
},
"COMM_ORIGIN": {
  "type": "object",
  "properties": {
    "COMM_APPLID": {
      "type": "string",
      "maxLength": 8
    },
    "COMM_USERID": {
      "type": "string",
      "maxLength": 8
    },
    "COMM_FACILITY_NAME": {
      "type": "string",
      "maxLength": 8
    }
  }
}

```

```

    },
    "COMM_NETWRK_ID":
        {
            "type": "string",
            "maxLength": 8
        },
    "COMM_FACILTYPE":
        {
            "type": "integer",
            "minimum": -999999999,
            "maximum": 999999999
        },
    "FILL_0":
        {
            "type": "string",
            "maxLength": 4
        }
    }
},
"COMM_SUCCESS":
    {
        "type": "string",
        "maxLength": 1
    },
"COMM_FAIL_CODE":
    {
        "type": "string",
        "maxLength": 1
    }
}
}}
}
}
}

```

## The CUSTOMER SERVICE Interface:

The Customer Service interface will be used in bank branches, by staff, who can traverse the line of queuing customers, to deal with most transactions that do not involve handling cash or cheques (these will still be dealt with by the Bank Teller). The customer services staff will have a tablet running the Customer Service interface, and they will be able to perform a range of functions e.g. changing a customer's name or address, closing an account, deal with account enquiries/balances etc.

For further details please refer to the Customer Service Interface User Guide.

## The associated restful Customer Service details:

The Customer Service Interface utilises a range of new z/OS Connect EE restful APIs. These restful APIs can also be driven independently of the Customer Service Interface and can, therefore, be integrated into other applications if desired.

### A. The Customer Service Customer Enquiry (CScustenq) details are:

```
"serviceName": "CScustenq",  
"serviceDescription": "Customer Service customer enquiry",  
"serviceProvider": "CICS-1.0",  
"serviceURL": "http://your-host-name:your-port-number  
/zosConnect/services/CScustenq"
```

### The Customer Service Customer Enquiry API:

There is one API available for the Customer Enquiry, and that is the *inqcustz* API. The details are as follows:

```
"name": "inqcustz",  
"version": "1.0.0",  
"description": "enquiry about a customer on the CScustenq service",  
"adminUrl": http://your-host-name:your-port-number/zosConnect/apis/inqcustz
```

### The URL to drive restful Customer Enquiry is:

<http://your-host-name:your-port-number/inqcustz/enquiry/10>

This URL enquires on customer 10 and issues an HTTP GET. No JSON request body is needed for this enquiry.

### The JSON *response* body returned is:

Here is the response from the above request:

```
{ "INQCUSTZ": {
  "INQCUST_NAME": "Dr William Q Price",
  "INQCUST_CREDIT_SCORE": 263,
  "INQCUST_INQ_SUCCESS": "Y",
  "INQCUST_ADDR": "19 Nutmeg Grove, Durham",
  "INQCUST_SCODE": "987654",
  "INQCUST_INQ_FAIL_CD": "0",
  "INQCUST_DOB": {
    "INQCUST_DOB_YYYY": 1936,
    "INQCUST_DOB_DD": 24,
    "INQCUST_DOB_MM": 9
  },
  "INQCUST_EYE": "CUST",
  "INQCUST_PCB_POINTER": "",
  "INQCUST_CUSTNO": 10,
  "INQCUST_CS_REVIEW_DT": {
    "INQCUST_CS_REVIEW_YYYY": 2022,
    "INQCUST_CS_REVIEW_DD": 9,
    "INQCUST_CS_REVIEW_MM": 2
  }
}
}}
```

In this instance the request was successful, denoted by the INQCUST\_INQ\_SUCCESS attribute containing 'Y'. On the response body, integer fields do not have leading zeros. We have intentionally chosen to utilise the full copybook on the JSON request and response, illustrating the full range of data which may be flowed. Should some attributes not be required, it is simple enough to change the mappings in zOS Connect EE.

#### Examples of the JSON *response* body returned if the customer enquiry cannot be made:

There are occasions when a customer enquiry might not be successful, for example:

1. Customer cannot be found: In this example, customer number 1234567890 does not exist. Here is the JSON response body returned under this circumstance:

```
{ "INQCUSTZ": {
  "INQCUST_NAME": "",
  "INQCUST_CREDIT_SCORE": 0,
  "INQCUST_INQ_SUCCESS": "N",
  "INQCUST_ADDR": "",
  "INQCUST_SCODE": "",
  "INQCUST_INQ_FAIL_CD": "1",
  "INQCUST_DOB": {
    "INQCUST_DOB_YYYY": 0,
    "INQCUST_DOB_DD": 0,
    "INQCUST_DOB_MM": 0
  },
  "INQCUST_EYE": "",
  "INQCUST_PCB_POINTER": "",
  "INQCUST_CUSTNO": 1234567890,
  "INQCUST_CS_REVIEW_DT": {
```

```

        "INQCUST_CS_REVIEW_YYYY": 0,
        "INQCUST_CS_REVIEW_DD": 0,
        "INQCUST_CS_REVIEW_MM": 0
    }
}

```

The failure is denoted by the INQCUST\_INQ\_SUCCESS being set to 'N' and the INQCUST\_INQ\_FAIL\_CD attribute being set to '1' ('1' = customer not found).

2. *A problem with theabend handling in the backend program:*

The JSON response body returned is:

```

{"INQCUSTZ": {
  "INQCUST_NAME": "",
  "INQCUST_CREDIT_SCORE": 0,
  "INQCUST_INQ_SUCCESS": "N",
  "INQCUST_ADDR": "",
  "INQCUST_SCODE": "",
  "INQCUST_INQ_FAIL_CD": "2",
  "INQCUST_DOB": {
    "INQCUST_DOB_YYYY": 0,
    "INQCUST_DOB_DD": 0,
    "INQCUST_DOB_MM": 0
  },
  "INQCUST_EYE": "",
  "INQCUST_PCB_POINTER": "",
  "INQCUST_CUSTNO": 123,
  "INQCUST_CS_REVIEW_DT": {
    "INQCUST_CS_REVIEW_YYYY": 0,
    "INQCUST_CS_REVIEW_DD": 0,
    "INQCUST_CS_REVIEW_MM": 0
  }
}
}

```

The failure is denoted by the INQCUST\_INQ\_SUCCESS being set to 'N' and the INQCUST\_INQ\_FAIL\_CD attribute being set to '2' ('2' = an issue was encountered processing the request in the backend program, specifically relating to theabend handling).

3. *An issue processing the VSAM data related to the enquiry:*

The JSON response body returned is:

```

{"INQCUSTZ": {
  "INQCUST_NAME": "",
  "INQCUST_CREDIT_SCORE": 0,
  "INQCUST_INQ_SUCCESS": "N",
  "INQCUST_ADDR": "",
  "INQCUST_SCODE": "",
  "INQCUST_INQ_FAIL_CD": "9",
  "INQCUST_DOB": {

```

```

        "INQCUST_DOB_YYYY": 0,
        "INQCUST_DOB_DD": 0,
        "INQCUST_DOB_MM": 0
    },
    "INQCUST_EYE": "",
    "INQCUST_PCB_POINTER": "",
    "INQCUST_CUSTNO": 124,
    "INQCUST_CS_REVIEW_DT": {
        "INQCUST_CS_REVIEW_YYYY": 0,
        "INQCUST_CS_REVIEW_DD": 0,
        "INQCUST_CS_REVIEW_MM": 0
    }
}
}}

```

The failure is denoted by the INQCUST\_INQ\_SUCCESS being set to 'N' and the INQCUST\_INQ\_FAIL\_CD attribute being set to '9' ('9' = an issue was encountered with VSAM, in the backend program, and the request cannot be fulfilled).

### What is being driven at the backend by this API?:

Backend program **INQCUST** is being driven by this API call utilising copybook **INQCUSTZ**. Note the copybook INQCUSTZ and INQCUST are identical except that in INQCUSTZ any POINTERS have been converted into PIC X(4), this is to allow zOS Connect EE to utilise the copybook correctly.

### Swagger details:

```

{
  "swagger": "2.0",
  "info": {
    "description": "enquiry about a customer on the CScusteng service",
    "version": "1.0.0",
    "title": "inqcustz"
  },
  "basePath": "/inqcustz",
  "schemes": [
    "https",
    "http"
  ],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {"/enquiry/{custno}": {"get": {

```

```

"tags": ["inqcustz"],
"operationId": "getCScustenq",
"parameters": [
    {
        "name": "Authorization",
        "in": "header",
        "required": false,
        "type": "string"
    },
    {
        "name": "custno",
        "in": "path",
        "required": true,
        "type": "string"
    },
    {
        "in": "body",
        "name": "getCScustenq_request",
        "description": "request body",
        "required": true,
        "schema": {"$ref": "#/definitions/getCScustenq_request"}
    }
],
"responses": {"200": {
    "description": "OK",
    "schema": {"$ref": "#/definitions/getCScustenq_response_200"}
}},
"definitions": {
    "getCScustenq_request": {
        "type": "object",
        "properties": {"INQCUSTZ": {
            "type": "object",
            "properties": {
                "INQCUST_EYE": {
                    "type": "string",
                    "maxLength": 4
                }
            }
        }
    }
}

```

```

},
"INQCUST_SCORE": {
  "type": "string",
  "maxLength": 6
},
"INQCUST_NAME": {
  "type": "string",
  "maxLength": 60
},
"INQCUST_ADDR": {
  "type": "string",
  "maxLength": 160
},
"INQCUST_DOB": {
  "type": "object",
  "properties": {
    "INQCUST_DOB_DD": {
      "type": "integer",
      "minimum": 0,
      "maximum": 99
    },
    "INQCUST_DOB_MM": {
      "type": "integer",
      "minimum": 0,
      "maximum": 99
    },
    "INQCUST_DOB_YYYY": {
      "type": "integer",
      "minimum": 0,
      "maximum": 9999
    }
  }
},
"INQCUST_CREDIT_SCORE": {
  "type": "integer",
  "minimum": 0,
  "maximum": 999
}

```



```

    },
    "INQCUST_CS_REVIEW_DT": {
        "type": "object",
        "properties": {
            "INQCUST_CS_REVIEW_DD": {
                "type": "integer",
                "minimum": 0,
                "maximum": 99
            },
            "INQCUST_CS_REVIEW_MM": {
                "type": "integer",
                "minimum": 0,
                "maximum": 99
            },
            "INQCUST_CS_REVIEW_YYYY": {
                "type": "integer",
                "minimum": 0,
                "maximum": 9999
            }
        }
    },
    "INQCUST_INQ_SUCCESS": {
        "type": "string",
        "maxLength": 1
    },
    "INQCUST_INQ_FAIL_CD": {
        "type": "string",
        "maxLength": 1
    },
    "INQCUST_PCB_POINTER": {
        "type": "string",
        "maxLength": 4
    }
}

}}

},
"getCScustenq_response_200": {

```

```

"type": "object",
"properties": {"INQCUSTZ":          {
  "type": "object",
  "properties":          {
    "INQCUST_EYE":          {
      "type": "string",
      "maxLength": 4
    },
    "INQCUST_SCODE":          {
      "type": "string",
      "maxLength": 6
    },
    "INQCUST_CUSTNO":          {
      "type": "integer",
      "minimum": 0,
      "maximum": 9999999999
    },
    "INQCUST_NAME":          {
      "type": "string",
      "maxLength": 60
    },
    "INQCUST_ADDR":          {
      "type": "string",
      "maxLength": 160
    },
    "INQCUST_DOB":          {
      "type": "object",
      "properties":          {
        "INQCUST_DOB_DD":          {
          "type": "integer",
          "minimum": 0,
          "maximum": 99
        },
        "INQCUST_DOB_MM":          {
          "type": "integer",
          "minimum": 0,
          "maximum": 99
        }
      }
    }
  }
}

```

```

    },
    "INQCUST_DOB_YYYY": {
      "type": "integer",
      "minimum": 0,
      "maximum": 9999
    }
  },
  "INQCUST_CREDIT_SCORE": {
    "type": "integer",
    "minimum": 0,
    "maximum": 999
  },
  "INQCUST_CS_REVIEW_DT": {
    "type": "object",
    "properties": {
      "INQCUST_CS_REVIEW_DD": {
        "type": "integer",
        "minimum": 0,
        "maximum": 99
      },
      "INQCUST_CS_REVIEW_MM": {
        "type": "integer",
        "minimum": 0,
        "maximum": 99
      },
      "INQCUST_CS_REVIEW_YYYY": {
        "type": "integer",
        "minimum": 0,
        "maximum": 9999
      }
    }
  },
  "INQCUST_INQ_SUCCESS": {
    "type": "string",
    "maxLength": 1
  },

```

```
    "INQCUST_INQ_FAIL_CD":                {
        "type": "string",
        "maxLength": 1
    },
    "INQCUST_PCB_POINTER":                 {
        "type": "string",
        "maxLength": 4
    }
}
}}
```

## B. The Customer Service Customer Update (CScustupd) details are:

```
"ServiceName": "CScustupd",
"ServiceDescription": "Customer Service customer update",
"ServiceProvider": "CICS-1.0",
"ServiceURL": "http://your-host-name:your-host-number
/zosConnect/services/CScustupd"
```

### The Customer Service Customer Update API:

There is one API available for the Customer Update, and that is the *updcust* API. The details are as follows:

```
"name": "updcust",
"version": "1.0.0",
"description": "Update a customer's details on the CScustupd Service",
"adminUrl": "http://your-host-name:your-host-
number/zosConnect/apis/updcust"
```

### The URL to drive restful Customer Update is:

<http://your-host-name:your-host-number/updcust/update>

This URL performs a customer update and issues an HTTP PUT and requires a JSON request body to be provided (containing either the amended Name, the amended Address or both) and a JSON response body is returned.

### An example JSON *request* body:

```
{
  "UPDCUST": {
    "COMM_EYE": " ",
    "COMM_SCODE": "987654",
    "COMM_CUSTNO": "0000000001",
    "COMM_NAME": "Mrs Buford G Morris-Symyth-Ping-Fred-Splat",
    "COMM_ADDR": "602A Maple Avenue, Fife",
    "COMM_DOB": 0,
    "COMM_CREDIT_SCORE": 0,
    "COMM_CS_REVIEW_DATE": 0,
    "COMM_UPD_SUCCESS": " ",
    "COMM_UPD_FAIL_CD": " "
  }
}
```

### The JSON *response* body returned is:

```
{"UPDCUST": {
  "COMM_ADDR": "602A Maple Avenue, Fife",
  "COMM_SCODE": "987654",
  "COMM_NAME": "Mrs Buford G Morris-Symyth-Ping-Fred-Splat",
  "COMM_UPD_SUCCESS": "Y",
  "COMM_EYE": "CUST",
```

```

"COMM_CS_REVIEW_DATE": 21012022,
"COMM_CUSTNO": "0000000001",
"COMM_CREDIT_SCORE": 516,
"COMM_UPD_FAIL_CD": "",
"COMM_DOB": 22021941
}}

```

In this instance the request was successful, denoted by the COMM\_UPD\_SUCCESS attribute containing 'Y'. On the response body, integer fields do not have leading zeros. We have intentionally chosen to utilise the full copybook on the JSON request and response, illustrating the full range of data which may be flowed. Should some attributes not be required, it is simple enough to change the mappings in zOS Connect EE.

**Examples of the JSON *response* body returned if the customer update cannot be made:**  
Occasions when a customer update might not be successful include:

1. *Customer cannot be found*: In this example, customer number 1234567890 does not exist, here is the JSON response body returned under this circumstance:

```

{"UPDCUST": {
  "COMM_ADDR": "70 Thames Avenue, Swindon, Wiltshire",
  "COMM_SCODE": "987654",
  "COMM_NAME": "Mrs Judith A Smith",
  "COMM_UPD_SUCCESS": "N",
  "COMM_EYE": "",
  "COMM_CS_REVIEW_DATE": 0,
  "COMM_CUSTNO": "1234567890",
  "COMM_CREDIT_SCORE": 0,
  "COMM_UPD_FAIL_CD": "1",
  "COMM_DOB": 0
}}

```

The failure is denoted by the COMM\_UPD\_SUCCESS being set to 'N' and the COMM\_UPD\_FAIL\_CD attribute being set to '1' ('1' = Customer not found).

2. *An issue processing the VSAM data related to the update*:

```

{"UPDCUST": {
  "COMM_ADDR": "702 Burt Street, Swindon, Wilts",
  "COMM_SCODE": "987654",
  "COMM_NAME": "Mrs Maude Grimms",
  "COMM_UPD_SUCCESS": "N",
  "COMM_EYE": "",
  "COMM_CS_REVIEW_DATE": 0,
  "COMM_CUSTNO": "0000000214",
  "COMM_CREDIT_SCORE": 0,
  "COMM_UPD_FAIL_CD": "2",
  "COMM_DOB": 0
}}

```

The failure is denoted by the COMM\_UPD\_SUCCESS being set to 'N' and the COMM\_UPD\_FAIL\_CD attribute being set to '2' ('2' = An issue was encountered with VSAM datastore, in the backend program, and the request cannot be fulfilled).

3. A problem with the Title:

```
{"UPDCUST": {  
  "COMM_ADDR": "70 Thames Avenue, Swindon, Wiltshire",  
  "COMM_SCODE": "987654",  
  "COMM_NAME": "Reverend Judith A Smith",  
  "COMM_UPD_SUCCESS": "N",  
  "COMM_EYE": "",  
  "COMM_CS_REVIEW_DATE": 0,  
  "COMM_CUSTNO": "0000000037",  
  "COMM_CREDIT_SCORE": 0,  
  "COMM_UPD_FAIL_CD": "T",  
  "COMM_DOB": 0  
}}
```

The failure is denoted by the COMM\_UPD\_SUCCESS being set to 'N' and the COMM\_UPD\_FAIL\_CD attribute being set to 'T' ('T' = An issue was encountered with the title used at the start of the name (which must be; Mr, Mrs, Miss, Ms, Dr, Professor, Drs, Lord, Sir, Lady) and the request cannot be fulfilled).

4. A VSAM update issue:

```
{"UPDCUST": {  
  "COMM_ADDR": "90 Severn Avenue, Swindon, Wilts",  
  "COMM_SCODE": "987654",  
  "COMM_NAME": "Mr James Watt",  
  "COMM_UPD_SUCCESS": "N",  
  "COMM_EYE": "",  
  "COMM_CS_REVIEW_DATE": 0,  
  "COMM_CUSTNO": "0000005426",  
  "COMM_CREDIT_SCORE": 0,  
  "COMM_UPD_FAIL_CD": "3",  
  "COMM_DOB": 0  
}}
```

The failure is denoted by the COMM\_UPD\_SUCCESS being set to 'N' and the COMM\_UPD\_FAIL\_CD attribute being set to '3' ('3' = VSAM failure; unable to update the details in the backend program, request cannot be fulfilled).

5. Customer Name and Address are both spaces:

```
{"UPDCUST": {  
  "COMM_ADDR": "",  
  "COMM_SCODE": "987654",  
  "COMM_NAME": " ",  
  "COMM_UPD_SUCCESS": "N",  
  "COMM_EYE": "",  
  "COMM_CS_REVIEW_DATE": 0,  
  "COMM_CUSTNO": "0000000000",  
  "COMM_CREDIT_SCORE": 0,  
  "COMM_UPD_FAIL_CD": "2",  
  "COMM_DOB": 0  
}}
```

```

"COMM_NAME": "",
"COMM_UPD_SUCCESS": "N",
"COMM_EYE": "",
"COMM_CS_REVIEW_DATE": 0,
"COMM_CUSTNO": "0000009971",
"COMM_CREDIT_SCORE": 0,
"COMM_UPD_FAIL_CD": "4",
"COMM_DOB": 0
}}

```

The failure is denoted by the COMM\_UPD\_SUCCESS being set to 'N' and the COMM\_UPD\_FAIL\_CD attribute being set to '4' ('4' = The Name and Address are both spaces or both start with a space, the request cannot be fulfilled). The rule that the name and address cannot both be set to spaces is enforced, to stop inadvertent update/corruption of the customer data.

#### Other information:

It is perfectly permissible to just update the name (and not supply the address) or to just update the address and not supply the name.

#### What is being driven at the backend by this API?:

Backend program **UPDCUST** is being driven by this API, the call utilises copybook **UPDCUST**.

#### Swagger details:

```

{
  "swagger": "2.0",
  "info": {
    "description": "Update a customer's details on the CScustupd Service",
    "version": "1.0.0",
    "title": "updcust"
  },
  "basePath": "/updcust",
  "schemes": [
    "https",
    "http"
  ],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/update": {
      "put": {
        "tags": ["updcust"],
        "operationId": "putCScustupd",
        "parameters": [
          {
            "in": "body",
            "name": "putCScustupd_request",
            "description": "request body",
            "required": true,

```



```

        "schema": {"$ref": "#/definitions/putCScustupd_request"}
    }],
    "responses": {"200": {
        "description": "OK",
        "schema": {"$ref": "#/definitions/putCScustupd_response_200"}
    }}
}],
"definitions": {
    "putCScustupd_request": {
        "type": "object",
        "properties": {"UPDCUST": {
            "type": "object",
            "properties": {
                "COMM_EYE": {
                    "type": "string",
                    "maxLength": 4
                },
                "COMM_SCORE": {
                    "type": "string",
                    "maxLength": 6
                },
                "COMM_CUSTNO": {
                    "type": "string",
                    "maxLength": 10
                },
                "COMM_NAME": {
                    "type": "string",
                    "maxLength": 60
                },
                "COMM_ADDR": {
                    "type": "string",
                    "maxLength": 160
                },
                "COMM_DOB": {
                    "type": "integer",
                    "minimum": 0,
                    "maximum": 99999999
                },
                "COMM_CREDIT_SCORE": {
                    "type": "integer",
                    "minimum": 0,
                    "maximum": 999
                },
                "COMM_CS_REVIEW_DATE": {
                    "type": "integer",
                    "minimum": 0,
                    "maximum": 99999999
                },
                "COMM_UPD_SUCCESS": {
                    "type": "string",
                    "maxLength": 1
                }
            }
        }
    }
}

```

```

    },
    "COMM_UPD_FAIL_CD": {
        "type": "string",
        "maxLength": 1
    }
}
}}
},
"putCScustupd_response_200": {
    "type": "object",
    "properties": {"UPDCUST": {
        "type": "object",
        "properties": {
            "COMM_EYE": {
                "type": "string",
                "maxLength": 4
            },
            "COMM_SCODE": {
                "type": "string",
                "maxLength": 6
            },
            "COMM_CUSTNO": {
                "type": "string",
                "maxLength": 10
            },
            "COMM_NAME": {
                "type": "string",
                "maxLength": 60
            },
            "COMM_ADDR": {
                "type": "string",
                "maxLength": 160
            },
            "COMM_DOB": {
                "type": "integer",
                "minimum": 0,
                "maximum": 99999999
            },
            "COMM_CREDIT_SCORE": {
                "type": "integer",
                "minimum": 0,
                "maximum": 999
            },
            "COMM_CS_REVIEW_DATE": {
                "type": "integer",
                "minimum": 0,
                "maximum": 99999999
            },
            "COMM_UPD_SUCCESS": {
                "type": "string",
                "maxLength": 1
            }
        }
    }
}

```

```
    },
    "COMM_UPD_FAIL_CD": {
      "type": "string",
      "maxLength": 1
    }
  }
}
}
```

### C. The Customer Service List All Customer's Accounts (CScustacc) details are:

```
"ServiceName": "CScustacc",
"ServiceDescription": "Customer Service customer's accounts",
"ServiceProvider": "CICS-1.0",
"ServiceURL": "http://your-host-name:your-port-number
/zosConnect/services/CScustacc"
```

#### The Customer Service List All Customer's Accounts API:

There is one API available for listing all of the customer's accounts, and that is the *inqaccz* API. The details are as follows:

```
"name": "inqaccz",
"version": "1.0.0",
"description": "enquire about an account on the CSaccenq service",
"adminUrl": http://your-host-name:your-port-
number/zosConnect/apis/inqaccz
```

#### The URL to drive restful List All Customer's Accounts Enquiry is:

<http://your-host-name:your-port-number/inqaccz/list/1>

This URL performs a list all customer's accounts and issues an HTTP PUT. No JSON request body is needed for this enquiry.

#### The JSON *response* body returned is:

Here is the response when the URL above is used:

```
{"INQACCCZ": {
  "COMM_FAIL_CODE": "0",
  "CUSTOMER_NUMBER": 1,
  "ACCOUNT_DETAILS": [
    {
      "COMM_AVAIL_BAL": 848150.91,
      "COMM_ACTUAL_BAL": 848150.91,
      "COMM_SCODE": "987654",
      "COMM_INT_RATE": 2.1,
      "COMM_EYE": "ACCT",
      "COMM_OPENED": 14041954,
      "COMM_CUSTNO": "0000000001",
      "COMM_NEXT_STMT_DT": 1082021,
      "COMM_ACC_TYPE": "ISA",
      "COMM_OVERDRAFT": 0,
      "COMM_ACCNO": 1,
      "COMM_LAST_STMT_DT": 1072021
    },
    {
      "COMM_AVAIL_BAL": 47974.08,
      "COMM_ACTUAL_BAL": 47974.08,
      "COMM_SCODE": "987654",
```

```

"COMM_INT_RATE": 1.75,
"COMM_EYE": "ACCT",
"COMM_OPENED": 18061996,
"COMM_CUSTNO": "0000000001",
"COMM_NEXT_STMT_DT": 1082021,
"COMM_ACC_TYPE": "SAVING",
"COMM_OVERDRAFT": 0,
"COMM_ACCNO": 2,
"COMM_LAST_STMT_DT": 1072021
},
{
"COMM_AVAIL_BAL": 252731.97,
"COMM_ACTUAL_BAL": 252731.97,
"COMM_SCODE": "987654",
"COMM_INT_RATE": 0,
"COMM_EYE": "ACCT",
"COMM_OPENED": 10031966,
"COMM_CUSTNO": "0000000001",
"COMM_NEXT_STMT_DT": 1082021,
"COMM_ACC_TYPE": "CURRENT",
"COMM_OVERDRAFT": 100,
"COMM_ACCNO": 3,
"COMM_LAST_STMT_DT": 1072021
},
{
"COMM_AVAIL_BAL": -673503.63,
"COMM_ACTUAL_BAL": -673503.63,
"COMM_SCODE": "987654",
"COMM_INT_RATE": 17.9,
"COMM_EYE": "ACCT",
"COMM_OPENED": 9111999,
"COMM_CUSTNO": "0000000001",
"COMM_NEXT_STMT_DT": 1082021,
"COMM_ACC_TYPE": "LOAN",
"COMM_OVERDRAFT": 0,
"COMM_ACCNO": 4,
"COMM_LAST_STMT_DT": 1072021
},
{
"COMM_AVAIL_BAL": -980076.1,
"COMM_ACTUAL_BAL": -980076.1,
"COMM_SCODE": "987654",
"COMM_INT_RATE": 5.25,
"COMM_EYE": "ACCT",
"COMM_OPENED": 6051999,
"COMM_CUSTNO": "0000000001",
"COMM_NEXT_STMT_DT": 1082021,
"COMM_ACC_TYPE": "MORTGAGE",
"COMM_OVERDRAFT": 0,
"COMM_ACCNO": 5,
"COMM_LAST_STMT_DT": 1072021
}

```

```

    }
  ],
  "COMM_PCB_POINTER": "",
  "CUSTOMER_FOUND": "Y",
  "COMM_SUCCESS": "Y"
}}

```

In this instance the request was successful, denoted by the COMM\_SUCCESS attribute containing 'Y'. Customer 1 had five accounts; an ISA account (account number 1), a SAVING account (acc no 2), a CURRENT account (acc no 3), a LOAN account (acc no 4) and a MORTGAGE account (acc no 5). On the response body, integer fields do not have leading zeros. We have intentionally chosen to utilise the full copybook on the JSON request and response, illustrating the full range of data which may be flowed. Should some attributes not be required, it is simple enough to change the mappings in zOS Connect EE.

**Examples of the JSON *response* body returned if the list accounts enquiry cannot be made:**  
Occasions when listing a customer's accounts might not be successful include:

1. *Customer cannot be found*: In this example customer number 1234567890 does not exist, here is the JSON response body returned under this circumstance:

```

{"INQACCCZ": {
  "COMM_FAIL_CODE": "1",
  "CUSTOMER_NUMBER": 1234567890,
  "COMM_PCB_POINTER": "",
  "CUSTOMER_FOUND": "N",
  "COMM_SUCCESS": "N"
}}

```

The failure is denoted by the COMM\_SUCCESS being set to 'N' and the COMM\_FAIL\_CODE set to '1'.

2. *Customer is found but has no accounts to list*: In this example customer number 10006 exists but does not have any related accounts. Here is the JSON response body returned under this circumstance:

```

{"INQACCCZ": {
  "COMM_FAIL_CODE": "0",
  "CUSTOMER_NUMBER": 10006,
  "COMM_PCB_POINTER": "",
  "CUSTOMER_FOUND": "Y",
  "COMM_SUCCESS": "Y"
}}

```

Note that this isn't a failure. The look up was successful, denoted by the COMM\_SUCCESS set to 'Y', the CUSTOMER\_FOUND set to 'Y' and the COMM\_FAIL\_CODE set to '0', as there were no associated accounts, none are returned.

### 3. An issue processing the Db2 account data:

```
{ "INQACCCZ": {  
    "COMM_FAIL_CODE": "3",  
    "CUSTOMER_NUMBER": 10008,  
    "COMM_PCB_POINTER": "",  
    "CUSTOMER_FOUND": "N",  
    "COMM_SUCCESS": "N"  
}}
```

The failure is denoted by the COMM\_SUCCESS being set to 'N' and the COMM\_FAIL\_CODE set to either '2', '3' or '4' ('2' = Cursor cannot be opened. '3' = An issue was encountered fetching the account data. '4' = An issue on the cursor close).

#### **What is being driven at the backend by this API?:**

Backend program **INQACCCU** is being driven by this API, which utilises copybook **INQACCCZ**. The copybook INQACCCZ and INQACCCU are identical except that in INQACCCZ any POINTERS have been converted into PIC X(4), this allows zOS Connect EE to utilise the copybook correctly.

#### **Swagger details:**

```
{  
  "swagger": "2.0",  
  "info": {  
    "description": "Enquire about the accounts belonging to a customer on  
the CScustacc service",  
    "version": "1.0.0",  
    "title": "inqacccz"  
  },  
  "basePath": "/inqacccz",  
  "schemes": [  
    "https",  
    "http"  
  ],  
  "consumes": ["application/json"],  
  "produces": ["application/json"],  
  "paths": {  
    "/list/{custno}": {  
      "get": {  
        "tags": ["inqacccz"],  
        "operationId": "getCScustacc",  
        "parameters": [  
          {  
            "name": "Authorization",  
            "in": "header",  
            "required": false,  
            "type": "string"  
          }  
        ]  
      }  
    }  
  }  
}
```

```

    },
    {
      "name": "custno",
      "in": "path",
      "required": true,
      "type": "string"
    },
    {
      "in": "body",
      "name": "getCScustacc_request",
      "description": "request body",
      "required": true,
      "schema": {"$ref": "#/definitions/getCScustacc_request"}
    }
  ],
  "responses": {"200": {
    "description": "OK",
    "schema": {"$ref": "#/definitions/getCScustacc_response_200"}
  }}
},
"definitions": {
  "getCScustacc_request": {
    "type": "object",
    "properties": {"INQACCCZ": {
      "type": "object",
      "properties": {
        "COMM_SUCCESS": {
          "type": "string",
          "maxLength": 1
        },
        "COMM_FAIL_CODE": {
          "type": "string",
          "maxLength": 1
        },
        "CUSTOMER_FOUND": {
          "type": "string",
          "maxLength": 1
        },
        "COMM_PCB_POINTER": {
          "type": "string",
          "maxLength": 4
        },
        "ACCOUNT_DETAILS": {
          "type": "array",
          "items": {
            "type": "object",
            "properties": {
              "COMM_EYE": {
                "type": "string",
                "maxLength": 4
              }
            }
          }
        }
      }
    }
  }
}

```



```

"COMM_CUSTNO": {
  "type": "string",
  "maxLength": 10
},
"COMM_SCODE": {
  "type": "string",
  "maxLength": 6
},
"COMM_ACCNO": {
  "type": "integer",
  "minimum": 0,
  "maximum": 99999999
},
"COMM_ACC_TYPE": {
  "type": "string",
  "maxLength": 8
},
"COMM_INT_RATE": {
  "type": "number",
  "format": "decimal",
  "minimum": 0,
  "maximum": 9999.99
},
"COMM_OPENED": {
  "type": "integer",
  "minimum": 0,
  "maximum": 99999999
},
"COMM_OVERDRAFT": {
  "type": "integer",
  "minimum": 0,
  "maximum": 99999999
},
"COMM_LAST_STMT_DT": {
  "type": "integer",
  "minimum": 0,
  "maximum": 99999999
},
"COMM_NEXT_STMT_DT": {
  "type": "integer",
  "minimum": 0,
  "maximum": 99999999
},
"COMM_AVAIL_BAL": {
  "type": "number",
  "format": "decimal",
  "minimum": -9.999999999999E9,
  "maximum": 9.999999999999E9
},
"COMM_ACTUAL_BAL": {
  "type": "number",

```

```

        "format": "decimal",
        "minimum": -9.999999999999E9,
        "maximum": 9.999999999999E9
    }
}
},
    "maxItems": 20,
    "minItems": 1
}
}
}}
},
"getCScustacc_response_200": {
    "type": "object",
    "properties": {"INQACCCZ": {
        "type": "object",
        "properties": {
            "CUSTOMER_NUMBER": {
                "type": "integer",
                "minimum": 0,
                "maximum": 9999999999
            },
            "COMM_SUCCESS": {
                "type": "string",
                "maxLength": 1
            },
            "COMM_FAIL_CODE": {
                "type": "string",
                "maxLength": 1
            },
            "CUSTOMER_FOUND": {
                "type": "string",
                "maxLength": 1
            },
            "COMM_PCB_POINTER": {
                "type": "string",
                "maxLength": 4
            },
            "ACCOUNT_DETAILS": {
                "type": "array",
                "items": {
                    "type": "object",
                    "properties": {
                        "COMM_EYE": {
                            "type": "string",
                            "maxLength": 4
                        },
                        "COMM_CUSTNO": {
                            "type": "string",
                            "maxLength": 10
                        }
                    }
                }
            },

```

```

"COMM_SCOPE": {
  "type": "string",
  "maxLength": 6
},
"COMM_ACCNO": {
  "type": "integer",
  "minimum": 0,
  "maximum": 99999999
},
"COMM_ACC_TYPE": {
  "type": "string",
  "maxLength": 8
},
"COMM_INT_RATE": {
  "type": "number",
  "format": "decimal",
  "minimum": 0,
  "maximum": 9999.99
},
"COMM_OPENED": {
  "type": "integer",
  "minimum": 0,
  "maximum": 99999999
},
"COMM_OVERDRAFT": {
  "type": "integer",
  "minimum": 0,
  "maximum": 99999999
},
"COMM_LAST_STMT_DT": {
  "type": "integer",
  "minimum": 0,
  "maximum": 99999999
},
"COMM_NEXT_STMT_DT": {
  "type": "integer",
  "minimum": 0,
  "maximum": 99999999
},
"COMM_AVAIL_BAL": {
  "type": "number",
  "format": "decimal",
  "minimum": -9.999999999999E9,
  "maximum": 9.999999999999E9
},
"COMM_ACTUAL_BAL": {
  "type": "number",
  "format": "decimal",
  "minimum": -9.999999999999E9,
  "maximum": 9.999999999999E9
}

```

```
        }  
      },  
      "maxItems": 20,  
      "minItems": 1  
    }  
  }  
}  
}
```

#### D. The Customer Service Customer Create (CScustcre) details are:

```
"ServiceName": "CScustcre",
"ServiceDescription": "Customer Service create customer",
"ServiceProvider": "CICS-1.0",
"ServiceURL": "http://your-host-name:your-port-number
/zosConnect/services/CScustcre"
```

#### The Customer Service Customer Creation API:

There is one API available for the Customer Create, and that is the **crecust** API. The details are as follows:

```
"name": "crecust",
"version": "1.0.0",
"description": "Creates a customer on the CScustcre Service",
"adminUrl": http://your-host-name:your-port-number/zosConnect/apis/crecust
```

#### The URL to drive restful Customer Creation is:

<http://your-host-name:your-port-number/crecust/insert>

This URL performs a customer creation by issuing an HTTP POST and requires a JSON request body to be provided (containing the Name, Address and Date of Birth) and a JSON response body is returned.

#### An example JSON *request* body:

```
{
  "CRECUST": {
    "COMM_EYECATCHER": "  ",
    "COMM_KEY": {
      "COMM_SORTCODE": 0,
      "COMM_NUMBER": 0
    },
    "COMM_NAME": "Mrs Fido Bloggs Mornington",
    "COMM_ADDRESS": "999 Midleton Road, York. YO1 1AA",
    "COMM_DATE_OF_BIRTH": 22071971,
    "COMM_CREDIT_SCORE": 0,
    "COMM_CS_REVIEW_DATE": 0,
    "COMM_SUCCESS": " ",
    "COMM_FAIL_CODE": " "
  }
}
```

#### The JSON *response* body returned is:

Here is the response from the above request:

```
{"CRECUST": {
  "COMM_ADDRESS": "999 Midleton Road, York. YO1 1AA",
```

```

"COMM_FAIL_CODE": "",
"COMM_NAME": "Mrs Fido Bloggs Mornington",
"COMM_KEY": {
  "COMM_SORTCODE": 987654,
  "COMM_NUMBER": 10007
},
"COMM_CS_REVIEW_DATE": 1022022,
"COMM_DATE_OF_BIRTH": 22071971,
"COMM_CREDIT_SCORE": 337,
"COMM_EYECATCHER": "CUST",
"COMM_SUCCESS": "Y"
}}

```

In this instance the request was successful and customer number 10007 was created, denoted by the COMM\_SUCCESS attribute containing 'Y'. Integer fields do not have leading zeros in the returned data. On the response body, integer fields do not have leading zeros. We have intentionally chosen to utilise the full copybook on the JSON request and response, illustrating the full range of data which may be flowed. Should some attributes not be required, it is simple enough to change the mappings in zOS Connect EE.

**Examples of the JSON *response* body returned if the customer creation cannot be made:**  
 Occasions when a customer creation might not be successful, include:

1. *Customer credit check failure:*

```

{"CRECUST": {
  "COMM_ADDRESS": "121 Acre Gap, Tring, Herts. TR9 7AP",
  "COMM_FAIL_CODE": "G",
  "COMM_NAME": "Mrs Freda Maurane",
  "COMM_KEY": {
    "COMM_SORTCODE": 0,
    "COMM_NUMBER": 0
  },
  "COMM_CS_REVIEW_DATE": 0,
  "COMM_DATE_OF_BIRTH": 0,
  "COMM_CREDIT_SCORE": 0,
  "COMM_EYECATCHER": "CUST",
  "COMM_SUCCESS": "N"
}}

```

The failure is denoted by the COMM\_SUCCESS being set to 'N' and the COMM\_FAIL\_CODE attribute being set to 'G' ('G' = issuing getting a credit score).

2. *An issue generating the new customer number:*

```

{"CRECUST": {
  "COMM_ADDRESS": "121 Acre Gap, Tring, Herts. TR9 7AP",
  "COMM_FAIL_CODE": "3",
  "COMM_NAME": "Mrs Freda Maurane",
  "COMM_KEY": {
    "COMM_SORTCODE": 0,

```

```

        "COMM_NUMBER": 0
    },
    "COMM_CS_REVIEW_DATE": 0,
    "COMM_DATE_OF_BIRTH": 0,
    "COMM_CREDIT_SCORE": 0,
    "COMM_EYECATCHER": "CUST",
    "COMM_SUCCESS": "N"
}}

```

The failure is denoted by the COMM\_SUCCESS being set to 'N' and the COMM\_FAIL\_CODE attribute being set to either '3', '4', or '5' ('3' = An issue with the enqueue. '4' = An issue updating the next customer number. '5' = An issue with the dequeue). In any of these cases the request cannot be fulfilled.

### 3. A problem with writing to a container:

```

{"CRECUST": {
    "COMM_ADDRESS": "121 Acre Gap, Tring, Herts. TR9 7AP",
    "COMM_FAIL_CODE": "A",
    "COMM_NAME": "Mrs Freda Maurane",
    "COMM_KEY": {
        "COMM_SORTCODE": 0,
        "COMM_NUMBER": 0
    },
    "COMM_CS_REVIEW_DATE": 0,
    "COMM_DATE_OF_BIRTH": 0,
    "COMM_CREDIT_SCORE": 0,
    "COMM_EYECATCHER": "CUST",
    "COMM_SUCCESS": "N"
}}

```

The failure is denoted by the COMM\_SUCCESS being set to 'N' and the COMM\_FAIL\_CODE attribute being set to 'A' ('A' = An issue writing information to a container and the request cannot be fulfilled).

### 4. A problem running an Asynchronous transaction (to generate a credit score):

```

{"CRECUST": {
    "COMM_ADDRESS": "121 Acre Gap, Tring, Herts. TR9 7AP",
    "COMM_FAIL_CODE": "B",
    "COMM_NAME": "Mrs Freda Maurane",
    "COMM_KEY": {
        "COMM_SORTCODE": 0,
        "COMM_NUMBER": 0
    },
    "COMM_CS_REVIEW_DATE": 0,
    "COMM_DATE_OF_BIRTH": 0,
    "COMM_CREDIT_SCORE": 0,

```

```

    "COMM_EYECATCHER": "CUST",
    "COMM_SUCCESS": "N"
  }}

```

The failure is denoted by the COMM\_SUCCESS being set to 'N' and the COMM\_FAIL\_CODE attribute being set to 'B' ('B' = Unable to successfully run an asynchronous transaction to calculate the credit score, request cannot be fulfilled).

5. Underlying infrastructure failure:

```

{"CRECUST": {
  "COMM_ADDRESS": "121 Acre Gap, Tring, Herts. TR9 7AP",
  "COMM_FAIL_CODE": "C",
  "COMM_NAME": "Mrs Freda Maurane",
  "COMM_KEY": {
    "COMM_SORTCODE": 0,
    "COMM_NUMBER": 0
  },
  "COMM_CS_REVIEW_DATE": 0,
  "COMM_DATE_OF_BIRTH": 0,
  "COMM_CREDIT_SCORE": 0,
  "COMM_EYECATCHER": "CUST",
  "COMM_SUCCESS": "N"
}}

```

The failure is denoted by the COMM\_SUCCESS being set to 'N' and the COMM\_FAIL\_CODE attribute being set to 'C', 'D', 'E', 'F', 'G', 'H' ('C' = None of the credit scoring agencies returned a credit score for this individual. 'D' = INVREQ condition flagged. 'E' = GET Container failed. 'F' = The FETCH condition abended. 'G' = A security issue was encountered. 'H' = Some other unspecified error occurred). In these cases, the request cannot be fulfilled.

6. Unable to record the new CUSTOMER on the datatstore:

```

{"CRECUST": {
  "COMM_ADDRESS": "121 Acre Gap, Tring, Herts. TR9 7AP",
  "COMM_FAIL_CODE": "1",
  "COMM_NAME": "Mrs Freda Maurane",
  "COMM_KEY": {
    "COMM_SORTCODE": 0,
    "COMM_NUMBER": 0
  },
  "COMM_CS_REVIEW_DATE": 0,
  "COMM_DATE_OF_BIRTH": 0,
  "COMM_CREDIT_SCORE": 0,
  "COMM_EYECATCHER": "CUST",
  "COMM_SUCCESS": "N"
}}

```



The failure is denoted by the COMM\_SUCCESS being set to 'N' and the COMM\_FAIL\_CODE attribute being set to '1' ('1' = Unable to record the new customer onto the CUSTOMER datastore). In this case, the request cannot be fulfilled.

7. The supplied DOB is invalid:

```
{ "CRECUST": {  
  "COMM_ADDRESS": "121 Acre Gap, Tring, Herts. TR9 7AP",  
  "COMM_FAIL_CODE": "O",  
  "COMM_NAME": "Mrs Freda Maurane",  
  "COMM_KEY": {  
    "COMM_SORTCODE": 0,  
    "COMM_NUMBER": 0  
  },  
  "COMM_CS_REVIEW_DATE": 0,  
  "COMM_DATE_OF_BIRTH": 0,  
  "COMM_CREDIT_SCORE": 0,  
  "COMM_EYECATCHER": "CUST",  
  "COMM_SUCCESS": "N"  
}}
```

The failure is denoted by the COMM\_SUCCESS being set to 'N' and the COMM\_FAIL\_CODE attribute being set to 'O' ('O' =The supplied Date Of Birth is invalid). In this case, the request cannot be fulfilled.

**Other information:**

The Name, Address and Date of Birth attributes should all contain valid data to drive this API correctly.

**What is being driven at the backend by this API?:**

Backend program **CRECUST** is being driven by this API, this utilises copybook **CRECUST**.

**Note of caution:**

The old saying “garbage in / garbage out” applies. The Restful Create Customer API should be used in accordance with the instructions as detailed above. Where possible it is recommended that the Create Customer API is integrated into an application, where validation checking of the JSON request and response bodies can be made. We perform such checking in the supplied Customer Services Interface. Scenarios outside of this, may need to perform additional validation checks to avoid data corruption.

### Swagger details:

```
{
  "swagger": "2.0",
  "info": {
    "description": "Creates a customer on the CScustcre Service",
    "version": "1.0.0",
    "title": "crecust"
  },
  "basePath": "/crecust",
  "schemes": [
    "https",
    "http"
  ],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {"/insert": {"post": {
    "tags": ["crecust"],
    "operationId": "postCScustcre",
    "parameters": [
      {
        "in": "body",
        "name": "postCScustcre_request",
        "description": "request body",
        "required": true,
        "schema": {"$ref": "#/definitions/postCScustcre_request"}
      }
    ],
    "responses": {"200": {
      "description": "OK",
      "schema": {"$ref": "#/definitions/postCScustcre_response_200"}
    }}
  }},
  "definitions": {
```

```

"postCScustcre_request":      {
  "type": "object",
  "properties": {"CRECUST":    {
    "type": "object",
    "properties":      {
      "COMM_EYECATCHER":      {
        "type": "string",
        "maxLength": 4
      },
      "COMM_KEY":            {
        "type": "object",
        "properties":        {
          "COMM_SORTCODE":    {
            "type": "integer",
            "minimum": 0,
            "maximum": 999999
          },
          "COMM_NUMBER":      {
            "type": "integer",
            "minimum": 0,
            "maximum": 9999999999
          }
        }
      },
      "COMM_NAME":          {
        "type": "string",
        "maxLength": 60
      },
      "COMM_ADDRESS":        {
        "type": "string",
        "maxLength": 160
      },

```

```

        "COMM_DATE_OF_BIRTH": {
            "type": "integer",
            "minimum": 0,
            "maximum": 99999999
        },
        "COMM_CREDIT_SCORE": {
            "type": "integer",
            "minimum": 0,
            "maximum": 999
        },
        "COMM_CS_REVIEW_DATE": {
            "type": "integer",
            "minimum": 0,
            "maximum": 99999999
        },
        "COMM_SUCCESS": {
            "type": "string",
            "maxLength": 1
        },
        "COMM_FAIL_CODE": {
            "type": "string",
            "maxLength": 1
        }
    }
}

    },
    "postCScustcre_response_200": {
        "type": "object",
        "properties": {
            "CRECUST": {
                "type": "object",
                "properties": {
                    "COMM_EYECATCHER": {

```

```

        "type": "string",
        "maxLength": 4
    },
    "COMM_KEY": {
        "type": "object",
        "properties": {
            "COMM_SORTCODE": {
                "type": "integer",
                "minimum": 0,
                "maximum": 999999
            },
            "COMM_NUMBER": {
                "type": "integer",
                "minimum": 0,
                "maximum": 9999999999
            }
        }
    },
    "COMM_NAME": {
        "type": "string",
        "maxLength": 60
    },
    "COMM_ADDRESS": {
        "type": "string",
        "maxLength": 160
    },
    "COMM_DATE_OF_BIRTH": {
        "type": "integer",
        "minimum": 0,
        "maximum": 99999999
    },
    "COMM_CREDIT_SCORE": {

```

```

        "type": "integer",

        "minimum": 0,

        "maximum": 999
    },

    "COMM_CS_REVIEW_DATE": {

        "type": "integer",

        "minimum": 0,

        "maximum": 99999999
    },

    "COMM_SUCCESS": {

        "type": "string",

        "maxLength": 1
    },

    "COMM_FAIL_CODE": {

        "type": "string",

        "maxLength": 1
    }
}

}}

}

}

}

```

## E. The Customer Service Delete a Customer (CScustdel) details are:

```
"ServiceName": "CScustdel",
"ServiceDescription": "Customer Service customer delete",
"ServiceProvider": "CICS-1.0",
"ServiceURL": "http://your-host-name:your-port-number
/zosConnect/services/CScustdel"
```

### The Customer Service Delete Customer API:

There is one API available for deleting customers, and that is the **delcus** API. The details are as follows:

```
"name": "delcus",
"version": "1.0.0",
"description": "Deletes a customer and their accounts on the CScustdel
Service",
"adminUrl": "http://your-host-name:your-port-number
/zosConnect/apis/delcus"
```

### The URL to drive restful Delete Customer is:

<http://your-host-name:your-port-number/delcus/remove/0000001471>

This URL performs a customer deletion via an HTTP DELETE. No JSON request body is needed for this enquiry.

### The JSON *response* body returned is:

Here is the response from the above request:

```
{"DELCUS": {
  "COMM_ADDR": "08 Holly Gate, Aberdeen",
  "COMM_SCODE": "987654",
  "COMM_NAME": "Ms Felicia B Evans",
  "COMM_DEL_SUCCESS": "Y",
  "COMM_EYE": "CUST",
  "COMM_CS_REVIEW_DATE": 28012022,
  "COMM_CUSTNO": "0000001471",
  "COMM_DEL_FAIL_CD": "",
  "COMM_CREDIT_SCORE": 255,
  "COMM_DOB": 11051992
}}
```

In this instance the delete customer request was successful, denoted by the COMM\_DEL\_SUCCESS attribute containing 'Y'. Customer 0000001471 was deleted. On the response body, integer fields do not have leading zeros. We have intentionally chosen to utilise the full copybook on the JSON request and response, illustrating the full range of data which may be flowed. Should some attributes not be required, it is simple enough to change the mappings in zOS Connect EE.

### Examples of the JSON *response* body returned if the delete request cannot be made:

Occasions when deleting a customer might not be successful, include:

1. *Customer cannot be found*: In this example customer number 1234567890 does not exist, here is the JSON response body returned under this circumstance:

```
{ "DELCUS": {  
  "COMM_ADDR": "",  
  "COMM_SCODE": "",  
  "COMM_NAME": "",  
  "COMM_DEL_SUCCESS": "N",  
  "COMM_EYE": "",  
  "COMM_CS_REVIEW_DATE": 0,  
  "COMM_CUSTNO": "1234567890",  
  "COMM_DEL_FAIL_CD": "1",  
  "COMM_CREDIT_SCORE": 0,  
  "COMM_DOB": 0  
}}
```

The failure is denoted by the COMM\_DEL\_SUCCESS being set to 'N' and the COMM\_DEL\_FAIL\_CD set to '1'.

#### What is being driven at the backend by this API?:

Backend program **DELCUS** is being driven by this API, the call utilises copybook **DELCUS**

#### Swagger details:

```
{  
  "swagger": "2.0",  
  "info": {  
    "description": "Deletes a customer and their accounts on the CScustdel  
Service",  
    "version": "1.0.0",  
    "title": "delcus"  
  },  
  "basePath": "/delcus",  
  "schemes": [  
    "https",  
    "http"  
  ],  
  "consumes": ["application/json"],  
  "produces": ["application/json"],  
  "paths": {  
    "/remove/{custno}": {  
      "delete": {  
        "tags": ["delcus"],  
        "operationId": "deleteCScustdel",  
        "parameters": [  
          {  
            "name": "Authorization",  
            "in": "header",  
            "required": false,  
            "type": "string"  
          }  
        ]  
      }  
    }  
  }  
}
```



```

    },
    {
      "name": "custno",
      "in": "path",
      "required": true,
      "type": "string",
      "maxLength": 10
    },
    {
      "in": "body",
      "name": "deleteCScustdel_request",
      "description": "request body",
      "required": true,
      "schema": {"$ref": "#/definitions/deleteCScustdel_request"}
    }
  ],
  "responses": {"200": {
    "description": "OK",
    "schema": {"$ref": "#/definitions/deleteCScustdel_response_200"}
  }}
},
"definitions": {
  "deleteCScustdel_request": {
    "type": "object",
    "properties": {"DELCUS": {
      "type": "object",
      "properties": {
        "COMM_EYE": {
          "type": "string",
          "maxLength": 4
        },
        "COMM_SCODE": {
          "type": "string",
          "maxLength": 6
        },
        "COMM_NAME": {
          "type": "string",
          "maxLength": 60
        },
        "COMM_ADDR": {
          "type": "string",
          "maxLength": 160
        },
        "COMM_DOB": {
          "type": "integer",
          "minimum": 0,
          "maximum": 99999999
        },
        "COMM_CREDIT_SCORE": {
          "type": "integer",
          "minimum": 0,

```

```

        "maximum": 999
    },
    "COMM_CS_REVIEW_DATE": {
        "type": "integer",
        "minimum": 0,
        "maximum": 99999999
    },
    "COMM_DEL_SUCCESS": {
        "type": "string",
        "maxLength": 1
    },
    "COMM_DEL_FAIL_CD": {
        "type": "string",
        "maxLength": 1
    }
}
}}
},
"deleteCScustdel_response_200": {
    "type": "object",
    "properties": {"DELCUS": {
        "type": "object",
        "properties": {
            "COMM_EYE": {
                "type": "string",
                "maxLength": 4
            },
            "COMM_SCODE": {
                "type": "string",
                "maxLength": 6
            },
            "COMM_CUSTNO": {
                "type": "string",
                "maxLength": 10
            },
            "COMM_NAME": {
                "type": "string",
                "maxLength": 60
            },
            "COMM_ADDR": {
                "type": "string",
                "maxLength": 160
            },
            "COMM_DOB": {
                "type": "integer",
                "minimum": 0,
                "maximum": 99999999
            },
            "COMM_CREDIT_SCORE": {
                "type": "integer",
                "minimum": 0,

```

```
        "maximum": 999
    },
    "COMM_CS_REVIEW_DATE": {
        "type": "integer",
        "minimum": 0,
        "maximum": 99999999
    },
    "COMM_DEL_SUCCESS": {
        "type": "string",
        "maxLength": 1
    },
    "COMM_DEL_FAIL_CD": {
        "type": "string",
        "maxLength": 1
    }
}

}}
```

## F. The Customer Service Account Enquiry (CSaccenq) details are:

```
"ServiceName": "CSaccenq",
"ServiceDescription": "Customer Service account enquiry",
"ServiceProvider": "CICS-1.0",
"ServiceURL": "http://your-host-name:your-port-number
/zosConnect/services/CSaccenq"
```

### The Customer Service Account Enquiry API:

There is one API available for the Account Enquiry, and that is the *inqaccz* API. The details are as follows:

```
"name": "inqaccz",
"version": "1.0.0",
"description": "enquire about an account on the CSaccenq service",
"adminUrl": "http://your-host-name:your-port-number
/zosConnect/apis/inqaccz"
```

### The URL to drive restful Account Enquiry is:

<http://your-host-name:your-port-number/inqaccz/enquiry/51>

This URL enquires on account 51 and issues an HTTP GET. No JSON request body is needed for this enquiry.

### The JSON *response* body returned is:

```
{ "INQACC_COMMAREA": {
  "INQACC_PCB1_POINTER": "",
  "INQACC_OVERDRAFT": 0,
  "INQACC_LAST_STMT_DT": 1072021,
  "INQACC_SCODE": 987654,
  "INQACC_ACTUAL_BAL": 514404.58,
  "INQACC_ACCNO": 51,
  "INQACC_OPENED": 10031921,
  "INQACC_CUSTNO": 16,
  "INQACC_ACC_TYPE": "SAVING",
  "INQACC_NEXT_STMT_DT": 1082021,
  "INQACC_AVAIL_BAL": 514404.58,
  "INQACC_EYE": "ACCT",
  "INQACC_SUCCESS": "Y",
  "INQACC_INT_RATE": 1.75
}}
```

In this instance the request was successful, denoted by the INQACC\_SUCCESS attribute containing 'Y'. On the response body, integer fields do not have leading zeros. We have intentionally chosen to utilise the full copybook on the JSON request and response, illustrating the full range of data which may be flowed. Should some attributes not be required, it is simple enough to change the mappings in zOS Connect EE.

### Examples of the JSON *response* body returned if the account enquiry cannot be made:

Occasions when an account enquiry might not be successful, include:

1. *Account cannot be found*: In this example, account number 12345678 does not exist, here is the JSON response body returned under this circumstance:

```
{ "INQACC_COMMAREA": {  
    "INQACC_PCB1_POINTER": "",  
    "INQACC_OVERDRAFT": 0,  
    "INQACC_LAST_STMT_DT": 0,  
    "INQACC_SCODE": 0,  
    "INQACC_ACTUAL_BAL": 0,  
    "INQACC_ACCNO": 12345678,  
    "INQACC_OPENED": 0,  
    "INQACC_CUSTNO": 0,  
    "INQACC_ACC_TYPE": "",  
    "INQACC_NEXT_STMT_DT": 0,  
    "INQACC_AVAIL_BAL": 0,  
    "INQACC_EYE": "",  
    "INQACC_SUCCESS": "N",  
    "INQACC_INT_RATE": 0  
  }  
}
```

The failure is denoted by the INQACC\_SUCCESS being set to 'N'.

#### What is being driven at the backend by this API?:

Backend program **INQACC** is being driven by this API call utilising copybook **INQAC CZ**. The copybook INQAC CZ and INQACC are identical except that in INQAC CZ any POINTERS have been converted into PIC X(4), this allows zOS Connect EE to utilise the copybook correctly.

#### Swagger details:

```
{  
  "swagger": "2.0",  
  "info": {  
    "description": "enquire about an account on the CSaccenq service",  
    "version": "1.0.0",  
    "title": "inqaccz"  
  },  
  "basePath": "/inqaccz",  
  "schemes": [  
    "https",  
    "http"  
  ],  
  "consumes": ["application/json"],  
  "produces": ["application/json"],  
  "paths": {  
    "/enquiry/{accno}": {  
      "get": {  
        "tags": ["inqaccz"],  
        "operationId": "getCSaccenq",  
        "parameters": [  

```

```

        {
            "name": "Authorization",
            "in": "header",
            "required": false,
            "type": "string"
        },
        {
            "name": "accno",
            "in": "path",
            "required": true,
            "type": "string"
        },
        {
            "in": "body",
            "name": "getCSaccenq_request",
            "description": "request body",
            "required": true,
            "schema": {"$ref": "#/definitions/getCSaccenq_request"}
        }
    ],
    "responses": {"200": {
        "description": "OK",
        "schema": {"$ref": "#/definitions/getCSaccenq_response_200"}
    }}
}},
"definitions": {
    "getCSaccenq_request": {
        "type": "object",
        "properties": {"INQACC_COMMAREA": {
            "type": "object",
            "properties": {
                "INQACC_EYE": {
                    "type": "string",
                    "maxLength": 4
                },
                "INQACC_CUSTNO": {
                    "type": "integer",
                    "minimum": 0,
                    "maximum": 9999999999
                },
                "INQACC_SCODE": {
                    "type": "integer",
                    "minimum": 0,
                    "maximum": 999999
                },
                "INQACC_ACC_TYPE": {
                    "type": "string",
                    "maxLength": 8
                },
                "INQACC_INT_RATE": {
                    "type": "number",

```

```

        "format": "decimal",
        "minimum": 0,
        "maximum": 9999.99
    },
    "INQACC_OPENED": {
        "type": "integer",
        "minimum": 0,
        "maximum": 99999999
    },
    "INQACC_OVERDRAFT": {
        "type": "integer",
        "minimum": 0,
        "maximum": 99999999
    },
    "INQACC_LAST_STMT_DT": {
        "type": "integer",
        "minimum": 0,
        "maximum": 99999999
    },
    "INQACC_NEXT_STMT_DT": {
        "type": "integer",
        "minimum": 0,
        "maximum": 99999999
    },
    "INQACC_AVAIL_BAL": {
        "type": "number",
        "format": "decimal",
        "minimum": -9.999999999999E9,
        "maximum": 9.999999999999E9
    },
    "INQACC_ACTUAL_BAL": {
        "type": "number",
        "format": "decimal",
        "minimum": -9.999999999999E9,
        "maximum": 9.999999999999E9
    },
    "INQACC_SUCCESS": {
        "type": "string",
        "maxLength": 1
    },
    "INQACC_PCB1_POINTER": {
        "type": "string",
        "maxLength": 4
    }
}

}}

},
"getCSaccenq_response_200": {
    "type": "object",
    "properties": {"INQACC_COMMAREA": {
        "type": "object",

```

```

"properties": {
  "INQACC_EYE": {
    "type": "string",
    "maxLength": 4
  },
  "INQACC_CUSTNO": {
    "type": "integer",
    "minimum": 0,
    "maximum": 9999999999
  },
  "INQACC_SCODE": {
    "type": "integer",
    "minimum": 0,
    "maximum": 999999
  },
  "INQACC_ACCNO": {
    "type": "integer",
    "minimum": 0,
    "maximum": 99999999
  },
  "INQACC_ACC_TYPE": {
    "type": "string",
    "maxLength": 8
  },
  "INQACC_INT_RATE": {
    "type": "number",
    "format": "decimal",
    "minimum": 0,
    "maximum": 9999.99
  },
  "INQACC_OPENED": {
    "type": "integer",
    "minimum": 0,
    "maximum": 99999999
  },
  "INQACC_OVERDRAFT": {
    "type": "integer",
    "minimum": 0,
    "maximum": 99999999
  },
  "INQACC_LAST_STMT_DT": {
    "type": "integer",
    "minimum": 0,
    "maximum": 99999999
  },
  "INQACC_NEXT_STMT_DT": {
    "type": "integer",
    "minimum": 0,
    "maximum": 99999999
  },
  "INQACC_AVAIL_BAL": {

```



```

        "type": "number",
        "format": "decimal",
        "minimum": -9.999999999999E9,
        "maximum": 9.999999999999E9
    },
    "INQACC_ACTUAL_BAL": {
        "type": "number",
        "format": "decimal",
        "minimum": -9.999999999999E9,
        "maximum": 9.999999999999E9
    },
    "INQACC_SUCCESS": {
        "type": "string",
        "maxLength": 1
    },
    "INQACC_PCBl_POINTER": {
        "type": "string",
        "maxLength": 4
    }
}

}}

}

}

}

```

## G. The Customer Service Account Update (CSaccupd) details are:

```
"ServiceName": "CSaccupd",
"ServiceDescription": "Customer Service account update",
"ServiceProvider": "CICS-1.0",
"ServiceURL": "http://your-host-name:your-port-number
/zosConnect/services/CSaccupd"
```

### The Customer Service Account Update API:

There is one API available for the Account Update, and that is the *updacc* API. The details are as follows:

```
"name": "updacc",
"version": "1.0.0",
"description": "Updates account information on the CSaccupd Service",
"adminUrl": "http://your-host-name:your-port-number
/zosConnect/apis/updacc"
```

### The URL to drive restful Account Update is:

<http://your-host-name:your-port-number/updacc/update>

This URL performs an account update and issues an HTTP PUT and requires a JSON request body to be provided (containing the account number, account type, interest rate, and overdraft limit) and a JSON response body is returned.

### An example JSON *request* body:

```
{
  "UPDACC": {
    "COMM_EYE": "    ",
    "COMM_CUSTNO": "0000000000",
    "COMM_SCODE": "000000",
    "COMM_ACCNO": 43,
    "COMM_ACC_TYPE": "LOAN    ",
    "COMM_INT_RATE": 66.91,
    "COMM_OPENED": 0,
    "COMM_OVERDRAFT": 1000,
    "COMM_LAST_STMT_DT": 0,
    "COMM_NEXT_STMT_DT": 0,
    "COMM_AVAIL_BAL": 0,
    "COMM_ACTUAL_BAL": 0,
    "COMM_SUCCESS": " "
  }
}
```

This request turns account 43 into a LOAN account, with an interest rate of 66.91%, and a 1000 overdraft limit.

### The JSON *response* body returned is:

```
{"UPDACC": {
  "COMM_OPENED": 1111980,
  "COMM_CUSTNO": "0000000013",
```

```

"COMM_ACC_TYPE": "LOAN",
"COMM_OVERDRAFT": 1000,
"COMM_ACCNO": 43,
"COMM_AVAIL_BAL": -642870.52,
"COMM_ACTUAL_BAL": -642870.52,
"COMM_SCODE": "987654",
"COMM_INT_RATE": 66.91,
"COMM_EYE": "ACCT",
"COMM_NEXT_STMT_DT": 1082021,
"COMM_LAST_STMT_DT": 1072021,
"COMM_SUCCESS": "Y"
}}

```

In this instance the request was successful, denoted by the COMM\_SUCCESS attribute containing 'Y'. On the response body, integer fields do not have leading zeros. We have intentionally chosen to utilise the full copybook on the JSON request and response, illustrating the full range of data which may be flowed. Should some attributes not be required, it is simple enough to change the mappings in zOS Connect EE.

**Examples of the JSON *response* body returned if the account update cannot be made:**  
Occasions when an account update might not be successful:

1. *Account cannot be found*: In this example account number 12345678 does not exist, here is the JSON response body returned under this circumstance:

```

{"UPDACC": {
  "COMM_OPENED": 0,
  "COMM_CUSTNO": "0000000000",
  "COMM_ACC_TYPE": "LOAN",
  "COMM_OVERDRAFT": 1000,
  "COMM_ACCNO": 12345678,
  "COMM_AVAIL_BAL": 0,
  "COMM_ACTUAL_BAL": 0,
  "COMM_SCODE": "987654",
  "COMM_INT_RATE": 66.91,
  "COMM_EYE": "",
  "COMM_NEXT_STMT_DT": 0,
  "COMM_LAST_STMT_DT": 0,
  "COMM_SUCCESS": "N"
}}

```

The failure is denoted by the COMM\_SUCCESS being set to 'N'.

2. *The account type is spaces or begins with a space*: Here is the JSON response body returned under this circumstance:

```

{"UPDACC": {
  "COMM_OPENED": 0,
  "COMM_CUSTNO": "0000000000",
  "COMM_ACC_TYPE": "",
  "COMM_OVERDRAFT": 1000,

```

```

"COMM_ACCNO": 73,
"COMM_AVAIL_BAL": 0,
"COMM_ACTUAL_BAL": 0,
"COMM_SCODE": "987654",
"COMM_INT_RATE": 66.91,
"COMM_EYE": "",
"COMM_NEXT_STMT_DT": 0,
"COMM_LAST_STMT_DT": 0,
"COMM_SUCCESS": "N"
}}

```

The failure is denoted by the COMM\_SUCCESS being set to 'N'.

3. Any other issues with the processing: Here is the JSON response body returned under this circumstance:

```

{"UPDACC": {
  "COMM_OPENED": 0,
  "COMM_CUSTNO": "0000000000",
  "COMM_ACC_TYPE": "ISA",
  "COMM_OVERDRAFT": 1000,
  "COMM_ACCNO": 6789,
  "COMM_AVAIL_BAL": 0,
  "COMM_ACTUAL_BAL": 0,
  "COMM_SCODE": "987654",
  "COMM_INT_RATE": 5.25,
  "COMM_EYE": "",
  "COMM_NEXT_STMT_DT": 0,
  "COMM_LAST_STMT_DT": 0,
  "COMM_SUCCESS": "N"
}}

```

If the update is unsuccessful for any other reasons you will also get the same response body (with COMM\_SUCCESS set to 'N').

#### Other information:

You should always supply an account number, an account type, the interest rate and the overdraft limit when undertaking an update account.

#### What is being driven at the backend by this API?:

Backend program **UPDACC** is being driven by this API, this utilises copybook **UPDACC**.

#### Note of caution:

The old "saying garbage in / garbage out" applies. The RestFul Account Update API should be used in accordance with the instructions as detailed above. Where possible it is

recommended that the Account Update API is integrated into an application, where validation checking of the JSON request and response bodies can be made. We perform such checking in the supplied Customer Services Interface. Scenarios outside of this, may need to perform additional validation checks to avoid data corruption.

### Swagger details:

```
{
  "swagger": "2.0",
  "info": {
    "description": "Updates account information on the CSaccupd Service",
    "version": "1.0.0",
    "title": "updacc"
  },
  "basePath": "/updacc",
  "schemes": [
    "https",
    "http"
  ],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {"/update": {"put": {
    "tags": ["updacc"],
    "operationId": "putCSaccupd",
    "parameters": [
      {
        "in": "body",
        "name": "putCSaccupd_request",
        "description": "request body",
        "required": true,
        "schema": {"$ref": "#/definitions/putCSaccupd_request"}
      }
    ],
    "responses": {"200": {
      "description": "OK",
      "schema": {"$ref": "#/definitions/putCSaccupd_response_200"}
    }}
  }},
  "definitions": {
    "putCSaccupd_request": {
      "type": "object",
      "properties": {"UPDACC": {
        "type": "object",
        "properties": {
          "COMM_EYE": {
            "type": "string",
            "maxLength": 4
          },
          "COMM_CUSTNO": {
            "type": "string",
            "maxLength": 10
          }
        }
      }
    }
  }
}
```

```

"COMM_CODE": {
  "type": "string",
  "maxLength": 6
},
"COMM_ACCNO": {
  "type": "integer",
  "minimum": 0,
  "maximum": 99999999
},
"COMM_ACC_TYPE": {
  "type": "string",
  "maxLength": 8
},
"COMM_INT_RATE": {
  "type": "number",
  "format": "decimal",
  "minimum": 0,
  "maximum": 9999.99
},
"COMM_OPENED": {
  "type": "integer",
  "minimum": 0,
  "maximum": 99999999
},
"COMM_OVERDRAFT": {
  "type": "integer",
  "minimum": 0,
  "maximum": 99999999
},
"COMM_LAST_STMT_DT": {
  "type": "integer",
  "minimum": 0,
  "maximum": 99999999
},
"COMM_NEXT_STMT_DT": {
  "type": "integer",
  "minimum": 0,
  "maximum": 99999999
},
"COMM_AVAIL_BAL": {
  "type": "number",
  "format": "decimal",
  "minimum": -9.999999999999E9,
  "maximum": 9.999999999999E9
},
"COMM_ACTUAL_BAL": {
  "type": "number",
  "format": "decimal",
  "minimum": -9.999999999999E9,
  "maximum": 9.999999999999E9
},

```

```

        "COMM_SUCCESS": {
            "type": "string",
            "maxLength": 1
        }
    }
}
}},
},
"putCSaccupd_response_200": {
    "type": "object",
    "properties": {"UPDACC": {
        "type": "object",
        "properties": {
            "COMM_EYE": {
                "type": "string",
                "maxLength": 4
            },
            "COMM_CUSTNO": {
                "type": "string",
                "maxLength": 10
            },
            "COMM_SCODE": {
                "type": "string",
                "maxLength": 6
            },
            "COMM_ACCNO": {
                "type": "integer",
                "minimum": 0,
                "maximum": 99999999
            },
            "COMM_ACC_TYPE": {
                "type": "string",
                "maxLength": 8
            },
            "COMM_INT_RATE": {
                "type": "number",
                "format": "decimal",
                "minimum": 0,
                "maximum": 9999.99
            },
            "COMM_OPENED": {
                "type": "integer",
                "minimum": 0,
                "maximum": 99999999
            },
            "COMM_OVERDRAFT": {
                "type": "integer",
                "minimum": 0,
                "maximum": 99999999
            },
            "COMM_LAST_STMT_DT": {
                "type": "integer",

```





## H. The Customer Service Account Create (CSacc cre) details are:

```
"ServiceName": "CSacc cre",
"ServiceDescription": "Customer Service account creation",
"ServiceProvider": "CICS-1.0",
"ServiceURL": "http://your-host-name:your-port-number
/zosConnect/services/CSacc cre"
```

### The Customer Service Account Creation API:

There is one API available for the Account Create, and that is the *creacc* API. The details are as follows:

```
"name": "creacc",
"version": "1.0.0",
"description": "Creates an account on the CSacc cre Service",
"adminUrl": "http://your-host-name:your-port-number
/zosConnect/apis/creacc"
```

### The URL to drive restful Account Creation is:

<http://your-host-name:your-port-number/creacc/insert>

This URL performs an account creation by issuing an HTTP POST and requires a JSON request body to be provided (containing the customer number, the account type, the interest rate, and the overdraft limit) and a JSON response body is returned.

### An example JSON *request* body:

```
{
  "CREACC": {
    "COMM_EYECATCHER": " ",
    "COMM_CUSTNO": 567,
    "COMM_KEY": {
      "COMM_SORTCODE": 0,
      "COMM_NUMBER": 0
    },
    "COMM_ACC_TYPE": "CURRENT ",
    "COMM_INT_RT": 9.25,
    "COMM_OPENED": 0,
    "COMM_OVERDR_LIM": 40,
    "COMM_LAST_STMT_DT": 0,
    "COMM_NEXT_STMT_DT": 0,
    "COMM_AVAIL_BAL": 0,
    "COMM_ACT_BAL": 0,
    "COMM_SUCCESS": " ",
    "COMM_FAIL_CODE": " "
  }
}
```

### The JSON *response* body returned is:

```
{"CREACC": {
  "COMM_OVERDR_LIM": 40,
```

```

"COMM_KEY": {
  "COMM_SORTCODE": 987654,
  "COMM_NUMBER": 29932
},
"COMM_OPENED": 26012022,
"COMM_CUSTNO": 567,
"COMM_ACC_TYPE": "CURRENT",
"COMM_INT_RT": 9.25,
"COMM_EYECATCHER": "ACCT",
"COMM_ACT_BAL": 0,
"COMM_AVAIL_BAL": 0,
"COMM_FAIL_CODE": "",
"COMM_NEXT_STMT_DT": 25022022,
"COMM_LAST_STMT_DT": 26012022,
"COMM_SUCCESS": "Y"
}}

```

In this instance, the request was successful and account number 29932 was created, denoted by the COMM\_SUCCESS attribute containing 'Y'. On the response body, integer fields do not have leading zeros. We have intentionally chosen to utilise the full copybook on the JSON request and response, illustrating the full range of data which may be flowed. Should some attributes not be required, it is simple enough to change the mappings in zOS Connect EE.

#### **Examples of the JSON *response* body returned if the account creation cannot be made:**

Occasions when an account creation might not be successful, include:

##### **1. Customer not found failure:**

```

{"CREACC": {
  "COMM_OVERDR_LIM": 40,
  "COMM_KEY": {
    "COMM_SORTCODE": 0,
    "COMM_NUMBER": 0
  },
  "COMM_OPENED": 0,
  "COMM_CUSTNO": 1234567890,
  "COMM_ACC_TYPE": "CURRENT",
  "COMM_INT_RT": 9.25,
  "COMM_EYECATCHER": "",
  "COMM_ACT_BAL": 0,
  "COMM_AVAIL_BAL": 0,
  "COMM_FAIL_CODE": "1",
  "COMM_NEXT_STMT_DT": 0,
  "COMM_LAST_STMT_DT": 0,
  "COMM_SUCCESS": "N"
}}

```

In this example, customer 1234567890 does not exist. The failure is denoted by the COMM\_SUCCESS being set to 'N' and the COMM\_FAIL\_CODE attribute being set to '1' ('1' = Customer not found).

2. **Bad account type supplied:**

```
{ "CREACC": {  
  "COMM_OVERDR_LIM": 40,  
  "COMM_KEY": {  
    "COMM_SORTCODE": 0,  
    "COMM_NUMBER": 0  
  },  
  "COMM_OPENED": 0,  
  "COMM_CUSTNO": 123,  
  "COMM_ACC_TYPE": "INVESTOR",  
  "COMM_INT_RT": 9.25,  
  "COMM_EYECATCHER": "",  
  "COMM_ACT_BAL": 0,  
  "COMM_AVAIL_BAL": 0,  
  "COMM_FAIL_CODE": "A",  
  "COMM_NEXT_STMT_DT": 0,  
  "COMM_LAST_STMT_DT": 0,  
  "COMM_SUCCESS": "N"  
}}
```

The failure is denoted by the COMM\_SUCCESS being set to 'N' and the COMM\_FAIL\_CODE attribute being set to 'A' ('A' = An invalid account type has been used. Allowable account types are 'ISA ', 'MORTGAGE', 'SAVING ', 'CURRENT ' or 'LOAN '). The request cannot be fulfilled.

3. **An issue with the account information being retrieved:**

```
{ "CREACC": {  
  "COMM_OVERDR_LIM": 40,  
  "COMM_KEY": {  
    "COMM_SORTCODE": 0,  
    "COMM_NUMBER": 0  
  },  
  "COMM_OPENED": 0,  
  "COMM_CUSTNO": 1818,  
  "COMM_ACC_TYPE": "MORTGAGE",  
  "COMM_INT_RT": 5.25,  
  "COMM_EYECATCHER": "",  
  "COMM_ACT_BAL": 0,  
  "COMM_AVAIL_BAL": 0,  
  "COMM_FAIL_CODE": "8",  
  "COMM_NEXT_STMT_DT": 0,  
  "COMM_LAST_STMT_DT": 0,  
  "COMM_SUCCESS": "N"  
}}
```

The failure is denoted by the COMM\_SUCCESS being set to 'N' and the COMM\_FAIL\_CODE attribute being set to either '8', or '9' ('8' = An error occurred

with the volume of account information retrieved. '9' = An error occurred counting the accounts). In any of these cases the request cannot be fulfilled.

4. *An internal problem generating the new account:*

```
{ "CREACC": {  
    "COMM_OVERDR_LIM": 40,  
    "COMM_KEY": {  
        "COMM_SORTCODE": 0,  
        "COMM_NUMBER": 0  
    },  
    "COMM_OPENED": 0,  
    "COMM_CUSTNO": 1818,  
    "COMM_ACC_TYPE": "MORTGAGE",  
    "COMM_INT_RT": 5.25,  
    "COMM_EYECATCHER": "",  
    "COMM_ACT_BAL": 0,  
    "COMM_AVAIL_BAL": 0,  
    "COMM_FAIL_CODE": "3",  
    "COMM_NEXT_STMT_DT": 0,  
    "COMM_LAST_STMT_DT": 0,  
    "COMM_SUCCESS": "N"  
}}
```

The failure is denoted by the COMM\_SUCCESS being set to 'N' and the COMM\_FAIL\_CODE attribute being set to '3', '5', '7' ('3' = An enqueue issue. '5' A dequeue issue. '7' = An issue with the INSERT). In any of these instances the request cannot be fulfilled.

**Other information:**

The customer number, account type, interest rate, and the overdraft limit attributes must all be provided to drive this API correctly.

**What is being driven at the backend by this API?:**

Backend program **CREACC** is being driven by this API, this utilises copybook **CREACC**.

**Note of caution:**

The old saying "garbage in / garbage out" applies. The RestFul Create Account API should be used in accordance with the instructions as detailed above. Where possible it is recommended that the Create Account API is integrated into an application, where validation checking of the JSON request and response bodies can be made. We perform such checking in the supplied Customer Services Interface. Scenarios outside of this, may need to perform additional validation checks to avoid data corruption.

### Swagger details:

```
{
  "swagger": "2.0",
  "info": {
    "description": "Creates an account on the CSacccre Service",
    "version": "1.0.0",
    "title": "creacc"
  },
  "basePath": "/creacc",
  "schemes": [
    "https",
    "http"
  ],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {"/insert": {"post": {
    "tags": ["creacc"],
    "operationId": "postCSacccre",
    "parameters": [
      {
        "name": "Authorization",
        "in": "header",
        "required": false,
        "type": "string"
      },
      {
        "in": "body",
        "name": "postCSacccre_request",
        "description": "request body",
        "required": true,
        "schema": {"$ref": "#/definitions/postCSacccre_request"}
      }
    ],
    "responses": {"200": {
      "description": "OK",
      "schema": {"$ref": "#/definitions/postCSacccre_response_200"}
    }}
  }},
  "definitions": {
    "postCSacccre_request": {
      "type": "object",
      "properties": {"CREACC": {
        "type": "object",
        "properties": {
          "COMM_EYECATCHER": {
            "type": "string",
            "maxLength": 4
          },
          "COMM_CUSTNO": {

```

```

        "type": "integer",
        "minimum": 0,
        "maximum": 9999999999
    },
    "COMM_KEY": {
        "type": "object",
        "properties": {
            "COMM_SORTCODE": {
                "type": "integer",
                "minimum": 0,
                "maximum": 999999
            },
            "COMM_NUMBER": {
                "type": "integer",
                "minimum": 0,
                "maximum": 99999999
            }
        }
    },
    "COMM_ACC_TYPE": {
        "type": "string",
        "maxLength": 8
    },
    "COMM_INT_RT": {
        "type": "number",
        "format": "decimal",
        "minimum": 0,
        "maximum": 9999.99
    },
    "COMM_OPENED": {
        "type": "integer",
        "minimum": 0,
        "maximum": 99999999
    },
    "COMM_OVERDR_LIM": {
        "type": "integer",
        "minimum": 0,
        "maximum": 99999999
    },
    "COMM_LAST_STMT_DT": {
        "type": "integer",
        "minimum": 0,
        "maximum": 99999999
    },
    "COMM_NEXT_STMT_DT": {
        "type": "integer",
        "minimum": 0,
        "maximum": 99999999
    },
    "COMM_AVAIL_BAL": {
        "type": "number",

```

```

        "format": "decimal",
        "minimum": -9.999999999999E9,
        "maximum": 9.999999999999E9
    },
    "COMM_ACT_BAL": {
        "type": "number",
        "format": "decimal",
        "minimum": -9.999999999999E9,
        "maximum": 9.999999999999E9
    },
    "COMM_SUCCESS": {
        "type": "string",
        "maxLength": 1
    },
    "COMM_FAIL_CODE": {
        "type": "string",
        "maxLength": 1
    }
}
}}
},
"postCSacccre_response_200": {
    "type": "object",
    "properties": {"CREACC": {
        "type": "object",
        "properties": {
            "COMM_EYECATCHER": {
                "type": "string",
                "maxLength": 4
            },
            "COMM_CUSTNO": {
                "type": "integer",
                "minimum": 0,
                "maximum": 9999999999
            },
            "COMM_KEY": {
                "type": "object",
                "properties": {
                    "COMM_SORTCODE": {
                        "type": "integer",
                        "minimum": 0,
                        "maximum": 999999
                    },
                    "COMM_NUMBER": {
                        "type": "integer",
                        "minimum": 0,
                        "maximum": 99999999
                    }
                }
            },
            "COMM_ACC_TYPE": {

```

```

        "type": "string",
        "maxLength": 8
    },
    "COMM_INT_RT": {
        "type": "number",
        "format": "decimal",
        "minimum": 0,
        "maximum": 9999.99
    },
    "COMM_OPENED": {
        "type": "integer",
        "minimum": 0,
        "maximum": 99999999
    },
    "COMM_OVERDR_LIM": {
        "type": "integer",
        "minimum": 0,
        "maximum": 99999999
    },
    "COMM_LAST_STMT_DT": {
        "type": "integer",
        "minimum": 0,
        "maximum": 99999999
    },
    "COMM_NEXT_STMT_DT": {
        "type": "integer",
        "minimum": 0,
        "maximum": 99999999
    },
    "COMM_AVAIL_BAL": {
        "type": "number",
        "format": "decimal",
        "minimum": -9.999999999999E9,
        "maximum": 9.999999999999E9
    },
    "COMM_ACT_BAL": {
        "type": "number",
        "format": "decimal",
        "minimum": -9.999999999999E9,
        "maximum": 9.999999999999E9
    },
    "COMM_SUCCESS": {
        "type": "string",
        "maxLength": 1
    },
    "COMM_FAIL_CODE": {
        "type": "string",
        "maxLength": 1
    }
}
}}

```



}  
}  
}

## I. The Customer Service Delete an Account (CSaccdel) details are:

```
"ServiceName": "CSaccdel",
"ServiceDescription": "Customer Services account delete",
"ServiceProvider": "CICS-1.0",
"ServiceURL": "http://your-host-name:your-port-number
/zosConnect/services/CSaccdel"
```

### The Customer Service Delete Account API:

There is one API available for deleting accounts, and that is the **delacc** API. The details are as follows:

```
"name": "delacc",
"version": "1.0.0",
"description": "Deletes an account on the CSaccdel Service",
"adminUrl": "http://your-host-name:your-port-number
/zosConnect/apis/delacc"
```

### The URL to drive restful Account deletion is:

<http://your-host-name:your-port-number/delacc/remove/559>

This URL performs an account delete and issues an HTTP DELETE. No JSON request body is needed for this.

### The JSON *response* body returned is:

```
{ "DELACC_COMMAREA": {
  "DELACC_SUCCESS": "",
  "DELACC_LAST_STMT_DT": 1072021,
  "DELACC_INT_RATE": 1.75,
  "DELACC_DEL_FAIL_CD": "",
  "DELACC_SCODE": "987654",
  "DELACC_DEL_PCB1": "",
  "DELACC_OPENED": 26112010,
  "DELACC_ACC_TYPE": "SAVING",
  "DELACC_NEXT_STMT_DT": 1082021,
  "DELACC_ACTUAL_BAL": 940129.93,
  "DELACC_AVAIL_BAL": 940129.93,
  "DELACC_CUSTNO": "0000000189",
  "DELACC_DEL_PCB3": "",
  "DELACC_DEL_PCB2": "",
  "DELACC_ACCNO": 559,
  "DELACC_OVERDRAFT": 0,
  "DELACC_FAIL_CD": "",
  "DELACC_EYE": "ACCT",
  "DELACC_DEL_APPLID": "",
  "DELACC_DEL_SUCCESS": "Y"
}}
```

In this example the delete account request was successful, denoted by the DELACC\_DEL\_SUCCESS attribute containing 'Y'. Account 559 was deleted. On the response body, integer fields do not have leading zeros. We have intentionally chosen to utilise the

full copybook on the JSON request and response, illustrating the full range of data which may be flowed. Should some attributes not be required, it is simple enough to change the mappings in zOS Connect EE.

**Examples of the JSON *response* body returned if the delete account request cannot be made:**

Occasions when an account deletion might not be successful, include:

1. *Account cannot be found*: In this example, account number 12345678 does not exist, here is the JSON response body returned under this circumstance:

```
{ "DELACC_COMMAREA": {  
    "DELACC_SUCCESS": "",  
    "DELACC_LAST_STMT_DT": 0,  
    "DELACC_INT_RATE": 0,  
    "DELACC_DEL_FAIL_CD": "1",  
    "DELACC_SCODE": "",  
    "DELACC_DEL_PCB1": "",  
    "DELACC_OPENED": 0,  
    "DELACC_ACC_TYPE": "",  
    "DELACC_NEXT_STMT_DT": 0,  
    "DELACC_ACTUAL_BAL": 0,  
    "DELACC_AVAIL_BAL": 0,  
    "DELACC_CUSTNO": "",  
    "DELACC_DEL_PCB3": "",  
    "DELACC_DEL_PCB2": "",  
    "DELACC_ACCNO": 12345678,  
    "DELACC_OVERDRAFT": 0,  
    "DELACC_FAIL_CD": "",  
    "DELACC_EYE": "",  
    "DELACC_DEL_APPLID": "",  
    "DELACC_DEL_SUCCESS": "N"  
  }  
}
```

The failure is denoted by the DELACC\_DEL\_SUCCESS = 'N' and the DELACC\_DEL\_FAIL\_CD = '1' (where '1' = Account not found).

2. *Account cannot be deleted from the datastore*: In this example, there is a technical problem and the account cannot be deleted from the datastore:

```
{ "DELACC_COMMAREA": {  
    "DELACC_SUCCESS": "",  
    "DELACC_LAST_STMT_DT": 0,  
    "DELACC_INT_RATE": 0,  
    "DELACC_DEL_FAIL_CD": "3",  
    "DELACC_SCODE": "",  
    "DELACC_DEL_PCB1": "",  
    "DELACC_OPENED": 0,  
    "DELACC_ACC_TYPE": "",  
    "DELACC_NEXT_STMT_DT": 0,  
    "DELACC_ACTUAL_BAL": 0,  
    "DELACC_DEL_SUCCESS": "N"  
  }  
}
```

```

        "DELACC_AVAIL_BAL": 0,
        "DELACC_CUSTNO": "",
        "DELACC_DEL_PCB3": "",
        "DELACC_DEL_PCB2": "",
        "DELACC_ACCNO": 3791,
        "DELACC_OVERDRAFT": 0,
        "DELACC_FAIL_CD": "",
        "DELACC_EYE": "",
        "DELACC_DEL_APPLID": "",
        "DELACC_DEL_SUCCESS": "N"
    }}

```

The failure is denoted by the DELACC\_DEL\_SUCCESS = 'N' and the DELACC\_DEL\_FAIL\_CD = '3' ('3' = Unable to delete).

### What is being driven at the backend by this API?:

Backend program **DELACC** is being driven by this API, the call utilises copybook **DELACCZ**. Note the copybook DELACCZ and DELACC are identical except that in DELACCZ any POINTERS have been converted into PIC X(4), allowing zOS Connect EE to utilise the copybook correctly.

### Swagger details:

```

{
  "swagger": "2.0",
  "info": {
    "description": "Deletes an account on the CSaccdel Service",
    "version": "1.0.0",
    "title": "delacc"
  },
  "basePath": "/delacc",
  "schemes": [
    "https",
    "http"
  ],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/remove/{accno}": {
      "delete": {
        "tags": ["delacc"],
        "operationId": "deleteCSaccdel",
        "parameters": [
          {
            "name": "Authorization",
            "in": "header",
            "required": false,
            "type": "string"
          },
          {

```

```

        "name": "accno",
        "in": "path",
        "required": true,
        "type": "string"
    },
    {
        "in": "body",
        "name": "deleteCSaccdel_request",
        "description": "request body",
        "required": true,
        "schema": {"$ref": "#/definitions/deleteCSaccdel_request"}
    }
],
"responses": {"200": {
    "description": "OK",
    "schema": {"$ref": "#/definitions/deleteCSaccdel_response_200"}
}}
}},
"definitions": {
    "deleteCSaccdel_request": {
        "type": "object",
        "properties": {"DELACC_COMMAREA": {
            "type": "object",
            "properties": {
                "DELACC_EYE": {
                    "type": "string",
                    "maxLength": 4
                },
                "DELACC_CUSTNO": {
                    "type": "string",
                    "maxLength": 10
                },
                "DELACC_SCODE": {
                    "type": "string",
                    "maxLength": 6
                },
                "DELACC_ACC_TYPE": {
                    "type": "string",
                    "maxLength": 8
                },
                "DELACC_INT_RATE": {
                    "type": "number",
                    "format": "decimal",
                    "minimum": 0,
                    "maximum": 9999.99
                },
                "DELACC_OPENED": {
                    "type": "integer",
                    "minimum": 0,
                    "maximum": 99999999
                }
            }
        }
    },

```

```

"DELACC_OVERDRAFT":                {
    "type": "integer",
    "minimum": 0,
    "maximum": 99999999
},
"DELACC_LAST_STMT_DT":              {
    "type": "integer",
    "minimum": 0,
    "maximum": 99999999
},
"DELACC_NEXT_STMT_DT":              {
    "type": "integer",
    "minimum": 0,
    "maximum": 99999999
},
"DELACC_AVAIL_BAL":                 {
    "type": "number",
    "format": "decimal",
    "minimum": -9.999999999999E9,
    "maximum": 9.999999999999E9
},
"DELACC_ACTUAL_BAL":                {
    "type": "number",
    "format": "decimal",
    "minimum": -9.999999999999E9,
    "maximum": 9.999999999999E9
},
"DELACC_SUCCESS":                   {
    "type": "string",
    "maxLength": 1
},
"DELACC_FAIL_CD":                   {
    "type": "string",
    "maxLength": 1
},
"DELACC_DEL_SUCCESS":                {
    "type": "string",
    "maxLength": 1
},
"DELACC_DEL_FAIL_CD":                {
    "type": "string",
    "maxLength": 1
},
"DELACC_DEL_APPLID":                 {
    "type": "string",
    "maxLength": 8
},
"DELACC_DEL_PCB1":                   {
    "type": "string",
    "maxLength": 4
},

```

```

        "DELACC_DEL_PCB2": {
            "type": "string",
            "maxLength": 4
        },
        "DELACC_DEL_PCB3": {
            "type": "string",
            "maxLength": 4
        }
    }
}
}},
},
"deleteCSaccdel_response_200": {
    "type": "object",
    "properties": {"DELACC_COMMAREA": {
        "type": "object",
        "properties": {
            "DELACC_EYE": {
                "type": "string",
                "maxLength": 4
            },
            "DELACC_CUSTNO": {
                "type": "string",
                "maxLength": 10
            },
            "DELACC_SCODE": {
                "type": "string",
                "maxLength": 6
            },
            "DELACC_ACCNO": {
                "type": "integer",
                "minimum": 0,
                "maximum": 99999999
            },
            "DELACC_ACC_TYPE": {
                "type": "string",
                "maxLength": 8
            },
            "DELACC_INT_RATE": {
                "type": "number",
                "format": "decimal",
                "minimum": 0,
                "maximum": 9999.99
            },
            "DELACC_OPENED": {
                "type": "integer",
                "minimum": 0,
                "maximum": 99999999
            },
            "DELACC_OVERDRAFT": {
                "type": "integer",
                "minimum": 0,

```

```

        "maximum": 99999999
    },
    "DELACC_LAST_STMT_DT": {
        "type": "integer",
        "minimum": 0,
        "maximum": 99999999
    },
    "DELACC_NEXT_STMT_DT": {
        "type": "integer",
        "minimum": 0,
        "maximum": 99999999
    },
    "DELACC_AVAIL_BAL": {
        "type": "number",
        "format": "decimal",
        "minimum": -9.999999999999E9,
        "maximum": 9.999999999999E9
    },
    "DELACC_ACTUAL_BAL": {
        "type": "number",
        "format": "decimal",
        "minimum": -9.999999999999E9,
        "maximum": 9.999999999999E9
    },
    "DELACC_SUCCESS": {
        "type": "string",
        "maxLength": 1
    },
    "DELACC_FAIL_CD": {
        "type": "string",
        "maxLength": 1
    },
    "DELACC_DEL_SUCCESS": {
        "type": "string",
        "maxLength": 1
    },
    "DELACC_DEL_FAIL_CD": {
        "type": "string",
        "maxLength": 1
    },
    "DELACC_DEL_APPLID": {
        "type": "string",
        "maxLength": 8
    },
    "DELACC_DEL_PCB1": {
        "type": "string",
        "maxLength": 4
    },
    "DELACC_DEL_PCB2": {
        "type": "string",
        "maxLength": 4
    }

```



```
    },
    "DELACC_DEL_PCB3": {
      "type": "string",
      "maxLength": 4
    }
  }
}
}
```