adiq

Documentation for 3DS 2.0 Implementation

Summary

Version control

Name	Date	Description	Version
Adiq	08/30/2020	Document Creation	1.0
Adiq	05/14/2021	Added the "Mandatory" column in the tables of the "Mandatory Fields and Validations in the ecommerce gateway" section	1.0.1
Adiq	08/18/2021	Adaptation of the mandatory fields; 3DS behavior when the customer also uses Anti-Fraud.	1.0.2
Adiq	07/12/2021	Handling the challenge response if the challenge process is cancelled.	1.0.3
Adiq	12/27/2022	Adjusted Payment.ProductType options in Domains table.	1.0.4
Adiq	05/31/2023	3ds library usage guidance	1.0.5
Adiq	09/28/2023	New version of “3ds-package” file	1.0.6
Adiq	05/10/2023	CyberSource Response Domain	1.0.7

Document link - https://developers.adiq.io/download/adiq-3ds-manual-1.0.7-en.pdf

Link developers - https://developers. adiq. io/manual/ecommerce#pagamento

1. Introduction

This document describes the step-by-step implementation of the 3DS 2.0 e-commerce authentication protocol. This document will detail the configuration for the **approval** environment.

The purpose of this document is sharing two examples of projects (with and without 3DS) so that the developers can situate themselves in the implementation of this protocol in their own e-commerce, in addition to mainly demonstrating all the steps and actions necessary to execute the implementation.

The base projects shared in the following topic are examples of ecommerce pages (a POC) that should be used as a reference for the implementation of 3DS 2.0. Naturally, each client that consumes this documentation will have their own e-commerce formats, file names, nomenclatures, methods, and classes with their own names.

Target Audience

The document has the following target audience:

- Developers of ADIQ clients who intend to implement the protocol.

2. About 3DS 2.0

In order to minimize the fraud rate without hindering the conversion rate, the means of payment industry developed 3DS 2. 0 is part of the payment flow, which allows banks to check whether an online purchase is fraudulent.

The solution comes to replace the 3DS 1. 0, which redirected the consumer to the bank page. In addition to repelling customers because of the fragmented experience, that form of authentication was still a problem for international companies, as its adoption varied widely between countries and banks.

To increase security around payments, a number of stricter rules have been created in Europe in recent years, and the 3D Secure 2.0 protocol is among them. The goal is to make the authentication process more dynamic and secure. 3DS is mandatory for debit transactions.

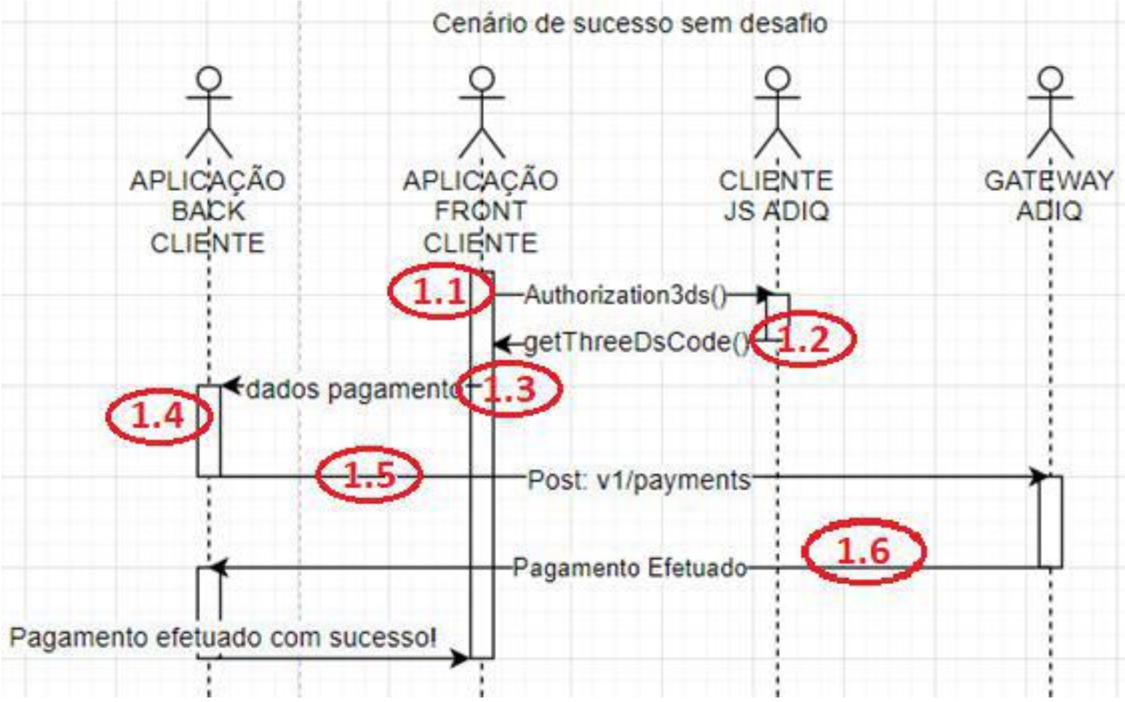
The service comes at additional cost.

How does 3DS work in practice?

3DS 2. 0 eliminates consumer redirect to the bank page. It works like this: the customer fills in your card information (credit or debit) and confirms the purchase. From this step, the payment processor gathers up to 100 different items of information about the transaction, which can be from the card number to the IP address, and sends this data to the card issuing bank with certified technology, ensuring a more fluid integration between websites and applications, always with the highest possible security.

3DS flowchart explained

Below we have a flowchart with a successful scenario, without challenge:



Picture breakdown:

1.1 At the time the customer who will pay for the purchase has already the card data filled in on the screen, the Authozation3ds **() method** must be called to start processing;

1.2 It will be necessary to generate the "code3ds" that is of the string type with size of 36 characters (the getThreeDsCode () method returns the "code3ds"), this 3ds code is important to be informed in the Gateway API "v1/payments" and "vl/payments/validate" ;

1.3 Capture the information below the Customer Browser, this data is in the Validations topic of this documentation, on pages 29 and 30.

- httpBrowserLanguage
- httpBrowserJavaEnabled
- httpBrowserJavaScriptEnabled
- httpBrowserColorDepth
- httpBrowserScreenHeight
- httpBrowserScreenWidth
- httpBrowserTimeDifference
- userAgentBrowserValue

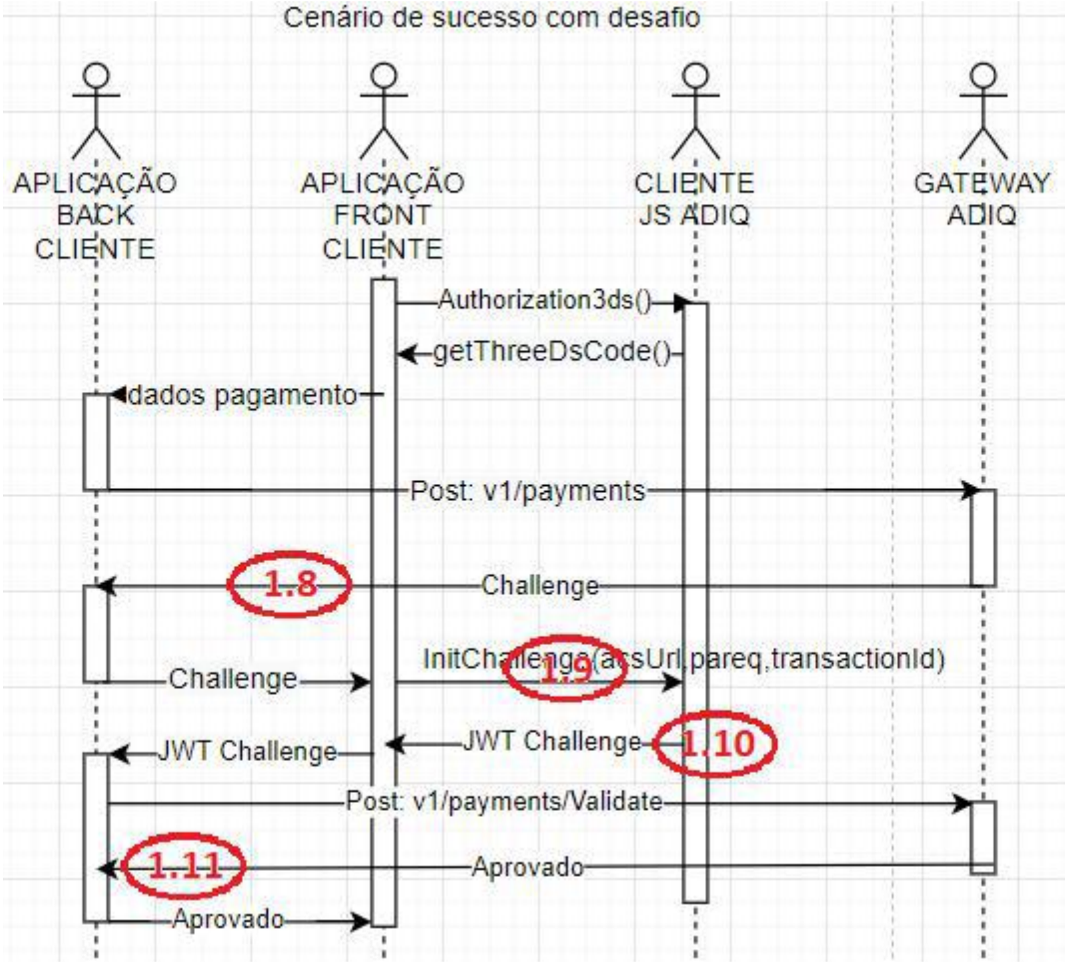
1.4 Send the "code3ds" and the fields about the browser mentioned in 1.3 step to the Backend;

1.5 In the “v1/payments” Gateway API, inform this data and see how to inform in the developer documentation (https://developers.adiq.io/manual/ecommerce);

1.6 Check in the response of "v1/payments" that it will return the data of the payment executed successfully.

```
{
“paymentAuthorization”: {
“returnCode”: “00”,
“description”: “Authorized”,
“paymentid”: “020080286103040952150000006201850000000000”,
“authorizationCode”: “043711”,
“orderNumber”: “0000000001”,
“expireAt”: “2019-09-24T13:20:52.8775511-03:00”,
“amount”: 1035,
“releaseAt”: “2019-09-24T13:20:52.877545-03:00”
},
```

Below, we have a flowchart with a successful scenario:



Description of additional scenarios above:

1.7 Follow the steps in items 1.1 to 1.5;

1.8 Check if the response of "v1/payments" returned the response "threeDs" as in the image below, if it returns with the status "challenge", then it must submit to the customer the challenge passing this data to the "adiq3ds" method;

```
                                “threeDs”: {
                                  “status”: “challenge”,
                                  “threeDsversion”: “2.1.9”,
                                  “acsUrl”: “https://mlerchantacsstag.cardinalcommerce.com/HerchantACSWeb/
                                  pareq.jsp?vaa=b&gold=AAAAAAA”, “authenticationTransactionld”:
                                  “WUcmoZFeccuB9fjhD8z0”,
                                  “pareq”:
“EnplUstuwjAQvPsrEOqxwnZCQoIWSzz64EBFaRtUbq7ZnomaBExSQb++dgilFbed2Yd3Zg3PsUacPKG
qNAqY4W4nP7CvrAft5YvKitUtklWN2HsaT4Jv1hYwHy5wk+AL9S4pcsE7rOMAPUFiRmgVy7wUINV2NH0QXT90uA+0gQQy1NOJYLzHmOe5ntcNGON
AjzSBXGZ4ma1Zaqqo8lIfBOuZiSdAoNKfIi7LTZ/Si06bI0Dpe80rG+2M3H2yFhGP0rdlkBZlL660Xh2t9rKm8cups<BUFtBYC1LFA5zGPNZ20JB
33X7jhld8wRkZhcRi6sW59fc8A1BYGOfGh4R5zb1lzFyKq0xVwcRB1bPCRHA/abI0dQYb39jo+K8+/je0qxK45nrC9ZzPWtxjev2xBjjeNaABphu
2ebA9Lm1ib69wd+Anm2pPg=”
                               }
```

1.9 In Front-end, startAdiq3ds.InitChallenge (**acsUrl**, pareq, authenticationTransactionld), after this, the client will receive a popup in the browser to answer the challenge;

1.10 After the Customer's response, they will receive a "validateToken" by the Call-back of the "validateChallengeCallback (page 14)", and this "validateToken" must be sent to the Back-end to process the API "v1/payments/validate" that will need the following fields: “code3ds” and “validateToken” . Thus, the Gateway will be able to process the payment;

1.11 Check the response from "v1/payments/validate" that will return the data of the payment executed. If the “paymentAuthorization” returns the contents of the attachment below, it means that the payment was successful.

```
{
“paymentAuthorization”: {
“returnCode”: “00”,
“description”: “Authorized”,
“paymentid”: “0200802861030409521500000062018500000000000”,
“authorizationCode”: “043711”,
“orderNumber”: “0000000001”,
“expireAt”: “2019-09-24T13:20:52.8775511-03:00”,
“amount”: 1035,
“releaseAt”: “2019-09-24T13:20:52.877545-03:00”
},
```

Next, we begin to check the links to download the projects that will serve as the basis for the implementation of 3DS 2. 0 as examples to be followed in your own e-commerce.

3DS behavior when the customer also uses Anti-Fraud:

If the customer purchases, in addition to 3DS, the Anti-Fraud service, the system behavior will be as follows:

- The 3DS service is called prior to Anti-fraud;
- Anti-fraud will only be called if the 3DS service has not been successful in credit transactions, that is, there is no need for an Anti-fraud check if the 3DS service has been successful.

- In debit transactions, the scenario is different: as 3DS is mandatory for debit transactions, if the 3DS service has not been successful, the transaction does not continue and, consequently, does not call Anti-fraud.
- If 3DS has returned a failure in a credit transaction, the Anti-Fraud service will be called and may prevent the transaction from occurring if the return is not successful.

3. Links and downloads

To implement 3DS, we have below some 3ds Javascript links.

Download link to the **"adiq-3ds-package" Approval Javascript Library file. adiq-3ds-package-1.0.1-hml-min.js"**:
https://developers.adiq.io/download/adiq-3ds-package-hml-1.0.1-min.js

Download link to the **"adiq-3ds-package-1.0.1-min.js"** Production Javascript Library file:
https://developers.adiq.io/download/adiq-3ds-package-1.0.1-min.js

To use the files, it is necessary to download them and replace the files with the most recent ones at each code update.
NOTE: It is highly not recommended to use through DNS referral due to unexpected errors during updates.

4. Procedures

Implementation of 3DS Protocol 2. 0

The implementation of the 3DS 2.0 protocol consists of linking a Javascript call to the button that executes a purchase. This button will presumably be on a page of the e-commerce website, in particular on the screen where the end customer executes the purchase after entering their credit/debit card.

Requirement

- Own a JQuery (version 3.3. 1 or above).

Process checkpoints

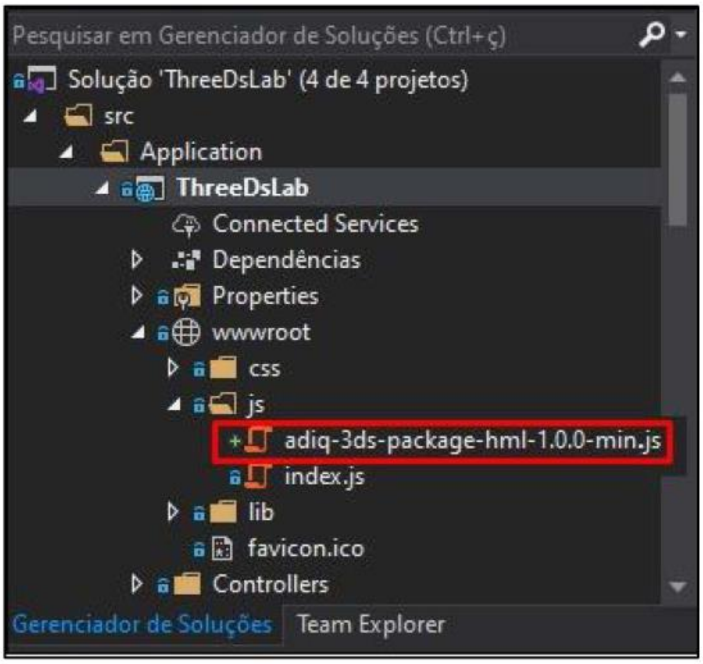
- Inject the JavaScript library;
- Create and update codes in the front end;
- Update codes in the back end;

Step 1 - Inject Javascript Library adiq-3ds-package-1.0.2-hml-min.js

To inject the JavaScript library into your directory where you load “jquery.min.js”, you must download the “adiq-3ds-package-1.0.1-**hml**-min.js” file that is in topic 2.

Note: In case of CORS-related errors when using the homologation package file, we recommend updating to version 1.0.2 of the file.

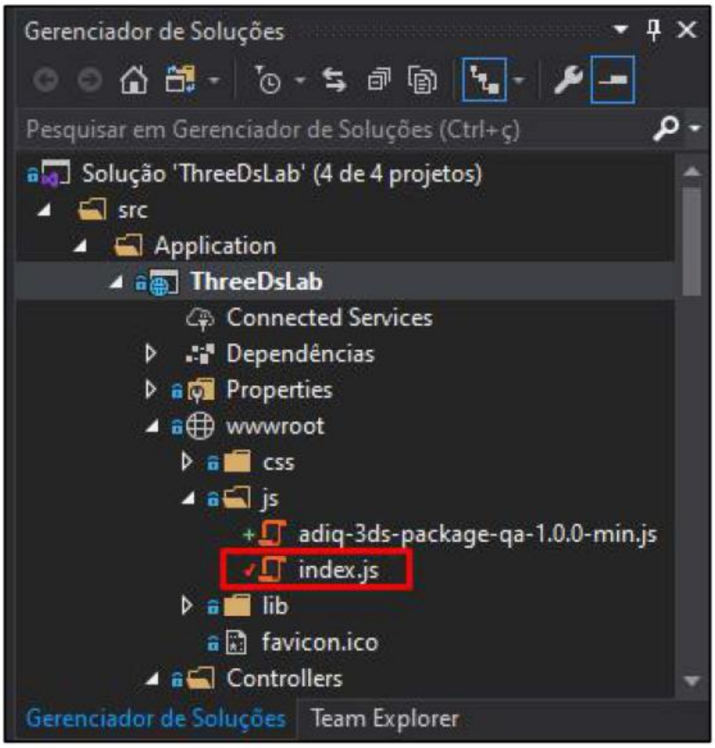
This file must be placed in your directory according to the attached image:



Step 2 - Development in the Front-End

The second step is to develop the code on the Front End, specifically on the e-commerce payment page where the “pay” button is available to end customers. It is the same screen where the credit/debit card is informed.

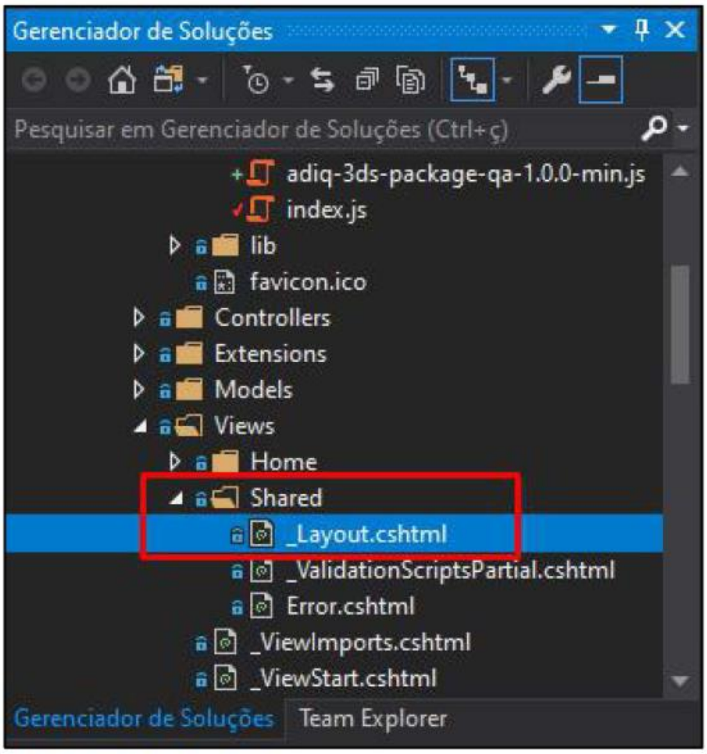
The following code snippet is an example of an **“index.js” file** that simulates a payment page. Pay attention only to the colorful excerpts of this page that are the excerpts referring to the implementation of 3DS, the other gray excerpts are codes expected on the e-commerce website (in this case, the basic example project):



```
Index.js

//link para o backend
window.backend_url = window.location.protocol + "//" + window.location.host + "/";

//link para capturar o IP
window.get_ip_url = "https://api.ipify.org/?format=json";

let threeDs;

//executando o pagamento
async function pagar() {
    console.log('pagar ...');

    threeDs = new Adiq3ds('ccnum', validateChallengeCallback);

    await threeDs.Authorization3ds();

    var code3ds = threeDs.getThreeDsCode();
    CallPayment(code3ds);

    //escondendo o botao pagar apos finalizar a operacao
    document.getElementById("pagarDiv").style.visibility = 'collapse';
}

function validateChallengeCallback(jwt, statusChallenge) {

    //status Approved/Cancelled
    console.log("Status Challenge: " + statusChallenge);
    if (statusChallenge == 'Cancelled') {
        alert('O desafio foi cancelado pelo usuário.');
        return;
    }

    //recebe a resposta do desafio
    //levar o jwt para o backend para ser informado no api Validate do gateway
    var validationData = getValidationData(jwt);

    var settings = {
        "async": true,
        "crossDomain": true,
        "url": window.backend_url + "Validation",
        "method": "POST",
        "headers": {
            "Content-Type": "application/json",
            "Accept": "*/*"
        },
        "processData": false,
        "data": JSON.stringify(validationData),
    };

    $.ajax(settings).done(function (response, status) {

        console.log("Status: " + status);

        console.log("Validation Response: " + JSON.stringify(response));

        ValidationResponse(response);

    }).fail(function (failResponse) {
        console.log("Problem during validation request.");
        logMessage('Problem during validation request.', true);
    });
}

//resposta da validacao
function ValidationResponse(validationResponse) {
    console.log("treatValidationResponse ... ");
    console.log(validationResponse);

    if (validationResponse.success) {
        window.datefin = performance.now();
        alert('Pagamento efetuado com sucesso! \nPaymentId: ' +
            validationResponse.paymentId +
            '\nAuthorizationCode: ' +
            validationResponse.authorizationCode +
            '\nAmount: ' +
            validationResponse.amount +
            '\ntime: ' +
            ((window.datefin - window.dateini) / 1000).toFixed(4) +
            ' seconds.');
    } else {
        window.datefin = performance.now();
        alert(validationResponse.description + '\ntime: ' + (window.datefin - window.dateini));
    }
}

//preparando o json do validation
function getValidationData(jwt) {
    var validationData;
    validationData = {
        code3DS: threeDs.getThreeDsCode(),
        validateToken: jwt
    };
    return validationData;
}

//processando gateway autorizacao
async function CallPayment(code3ds) {
    let payload = getPaymentData(code3ds);

    let r = await $.ajax({
        type: 'POST',
        dataType: 'json',
        contentType: 'application/json',
        url: window.payment_url,
        data: JSON.stringify(payload)
    });
    treatEnrollmentResponse(r);
}

//tratando a resposta da autorizacao gateway
function treatEnrollmentResponse(objEnrollment) {

    console.log("3DS protocol version: " + objEnrollment.ThreeDsVersion);
    console.log(objEnrollment);
    switch (objEnrollment.status) {
    case 2://falha na operacao
        console.log("falhou.");
        alert('Falha:' + objEnrollment.description);
        break;
    case 3://pagamento efetuado
        console.log("pago com sucesso!");
        window.datefin = performance.now();
        alert('Pagamento efetuado com sucesso! \nPaymentId: ' + objEnrollment.paymentId +
'\nAuthorizationCode: ' + objEnrollment.authorizationCode + '\nAmount: ' + objEnrollment.amount
+ '\ntime: ' + ((window.datefin - window.dateini) / 1000).toFixed(4) + ' seconds.');
        ShowReset();
        break;
    default://desafio
        console.log("Step-up.");
        if (confirm("Responda o desafio?")) {
            threeDs.InitChallenge(objEnrollment.acsUrl, objEnrollment.pareq,
objEnrollment.authenticationTransactionId);
        } else {
            console.log("Step-up canceled.");
            alert('Não foi autorizado o pagamento!');
        }
        break;
    }
}

//mostrando o botao Reset
```

```
function ShowReset() {
    document.getElementById("resetDiv").style.visibility = 'visible';
    document.getElementById("resetBtn").disable = false;
    document.getElementById("resetBtn").onclick = reset;
}

//capturando o ip do cliente
function GetIpAddress() {
    $.getJSON(window.get_ip_url,
        function (data) {
            console.log("IP Address: " + data.ip);
            document.getElementById('ipAddress').setAttribute('value', data.ip);
        });
}

//montagem do json para autorizacao
function getPaymentData(code3ds) {

    var paymentData;

    mountDeviceFingerprint();

    paymentData = {
        pagamento: {
            tipoTransacao: document.getElementById("overridePaymentMethod").value,
            codigoMoeda: document.getElementById("currency").value,
            valorTotal: document.getElementById("totalAmount").value
        },
        cartao: {
            numero: document.getElementById("ccnum").value,
            nomeProprietario: document.getElementById("cname").value,
            cvv: document.getElementById("cvv").value,
            bandeira: document.getElementById("type").value,
            mesVenc: document.getElementById("expmonth").value,
            anoVenc: document.getElementById("expyear").value,
        },
        vendedor: {
            code3DS: code3ds
        },
        aparelho: {
            ipAddress: document.getElementById("ipAddress").value,
            httpBrowserLanguage: document.getElementById("httpBrowserLanguage").value,
            httpBrowserJavaEnabled: document.getElementById("httpBrowserJavaEnabled").value,
            httpBrowserJavaScriptEnabled:
document.getElementById("httpBrowserJavaScriptEnabled").value,
            httpBrowserColorDepth: document.getElementById("httpBrowserColorDepth").value,
            httpBrowserScreenHeight: document.getElementById("httpBrowserScreenHeight").value,
            httpBrowserScreenWidth: document.getElementById("httpBrowserScreenWidth").value,
            httpBrowserTimeDifference:
document.getElementById("httpBrowserTimeDifference").value,
            userAgentBrowserValue: navigator.userAgent,
        }
    };
    return paymentData;
}

function mountDeviceFingerprint() {
    console.log("Getting 3DS devicefingerprint information.");

    var httpBrowserColorDepth = screen.colorDepth;
    document.getElementById('httpBrowserColorDepth').setAttribute('value',
httpBrowserColorDepth);
    console.log("httpBrowserColorDepth: " + httpBrowserColorDepth);
    var httpBrowserJavaEnabled = "";
    if (navigator.javaEnabled() == true) {
        httpBrowserJavaEnabled = "Y";
    }
    else {
        httpBrowserJavaEnabled = "N";
    }
    document.getElementById('httpBrowserJavaEnabled').setAttribute('value',
httpBrowserJavaEnabled);
    console.log("httpBrowserJavaEnabled: " + httpBrowserJavaEnabled);

    var httpBrowserJavaScriptEnabled = true;
    document.getElementById('httpBrowserJavaScriptEnabled').setAttribute('value',
httpBrowserJavaScriptEnabled);
    console.log("httpBrowserJavaScriptEnabled: " + httpBrowserJavaScriptEnabled);

    var httpBrowserLanguage = navigator.language || navigator.userLanguage;
    document.getElementById('httpBrowserLanguage').setAttribute('value', httpBrowserLanguage);
    console.log("httpBrowserLanguage: " + httpBrowserLanguage);

    var httpBrowserScreenHeight = window.innerHeight;
    document.getElementById('httpBrowserScreenHeight').setAttribute('value',
httpBrowserScreenHeight);
    console.log("httpBrowserScreenHeight: " + httpBrowserScreenHeight);

    var httpBrowserScreenWidth = window.innerWidth;
    document.getElementById('httpBrowserScreenWidth').setAttribute('value',
httpBrowserScreenWidth);
    console.log("httpBrowserScreenWidth: " + httpBrowserScreenWidth);

    var httpBrowserTimeDifference = new Date().getTimezoneOffset();
    document.getElementById('httpBrowserTimeDifference').setAttribute('value',
httpBrowserTimeDifference);
    console.log("httpBrowserTimeDifference: " + httpBrowserTimeDifference);
}

//reload na pagina para refazer o teste
function reset() {
    console.log("reset page....");
    location.reload();
}

//configuracao inicial
window.onload = function () {
    console.log("iniciando java....");
    document.getElementById('pagarBtn').onclick = pagar;
    GetIpAddress();
};
```

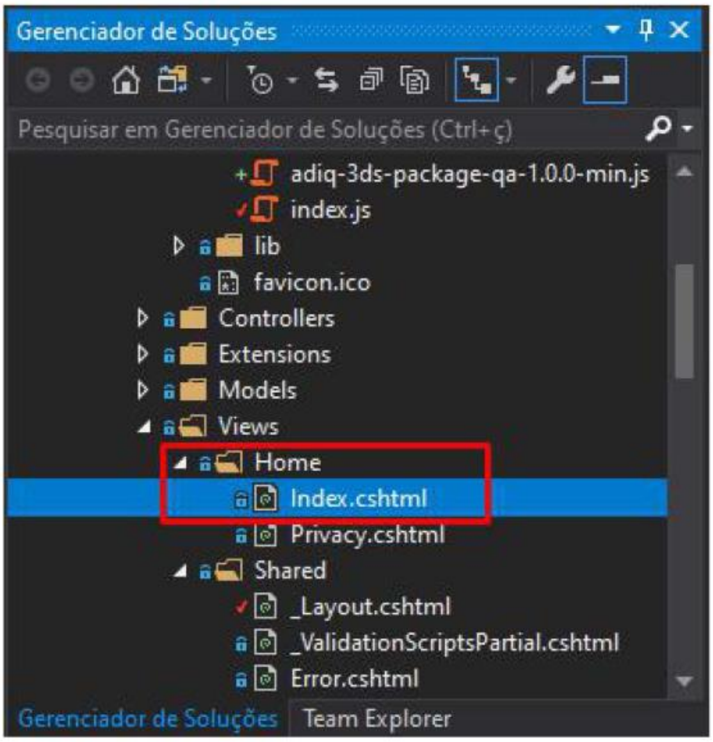
The following code snippet is an example of a "_Layout.cshtml" file that simulates where the 3ds **adiq-3ds-package-hml-1.0.1-min.js**, attach this file**.** Pay attention only to the colorful excerpts of this page that are the excerpts referring to the implementation of 3DS, the other gray excerpts are codes expected on the e-commerce website (in this case, the basic example project):



_Layout.cshtml

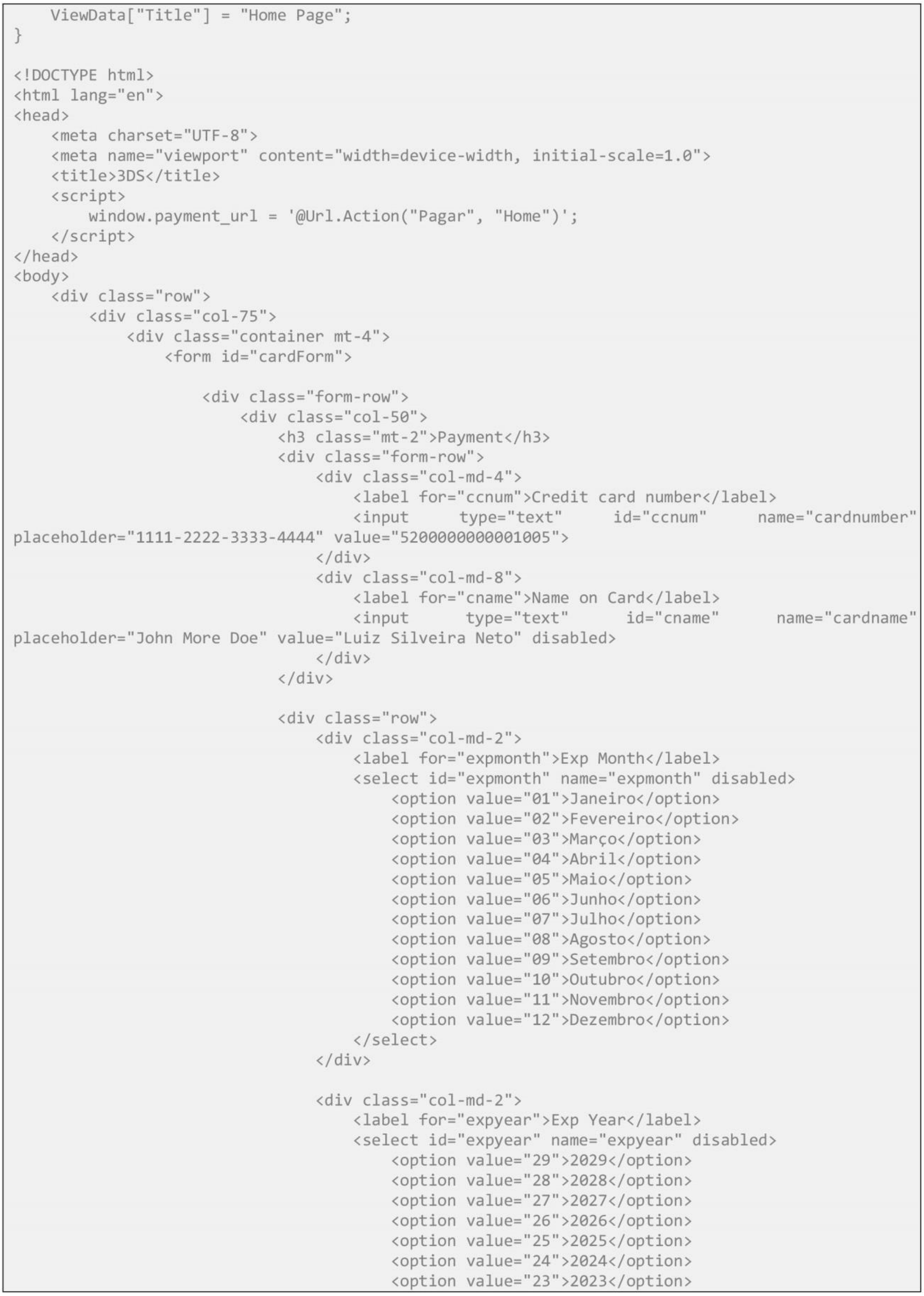
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>@ViewData["Title"] - ThreeDsLab</title>
    <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
    <link rel="stylesheet" href="~/css/site.css" />
    <link rel="stylesheet" href="~/css/pagamento.css" />
</head>
<body>
    <header>
        <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-
bottom box-shadow mb-3">
            <div class="container">
                <a class="navbar-brand" asp-area="" asp-controller="Home" asp-
action="Index">ThreeDsLab</a>
                <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target=".navbar-collapse" aria-controls="navbarSupportedContent"
                        aria-expanded="false" aria-label="Toggle navigation">
                    <span class="navbar-toggler-icon"></span>
                </button>
                <div class="navbar-collapse collapse d-sm-inline-flex flex-sm-row-reverse">
                    <ul class="navbar-nav flex-grow-1">
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-area="" asp-controller="Home"
asp-action="Index">Home</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-area="" asp-controller="Home"
asp-action="Privacy">Privacy</a>
                        </li>
                    </ul>
                </div>
            </div>
        </nav>
    </header>
    <div class="container">
        <main role="main" class="pb-3">
            @RenderBody()
        </main>
    </div>

    <footer class="border-top footer text-muted">
        <div class="container">
            &copy; 2020 - ThreeDsLab - <a asp-area="" asp-controller="Home" asp-
action="Privacy">Privacy</a>
        </div>
    </footer>
    <script src="~/lib/jquery/dist/jquery.min.js"></script>
    <script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
    <script src="~/js/adiq-3ds-package-hml-1.0.1-min.js" asp-append-version="true"></script>
    @RenderSection("Scripts", required: false)
</body>
</html>
```

The following code snippet is an example of a **"index.cshtml"** file that simulates is the payment HTML page. Pay attention only to the colorful excerpts of this page that are the excerpts referring to the implementation of 3DS, the other gray excerpts are codes expected on the e-commerce website (in this case, the basic example project):

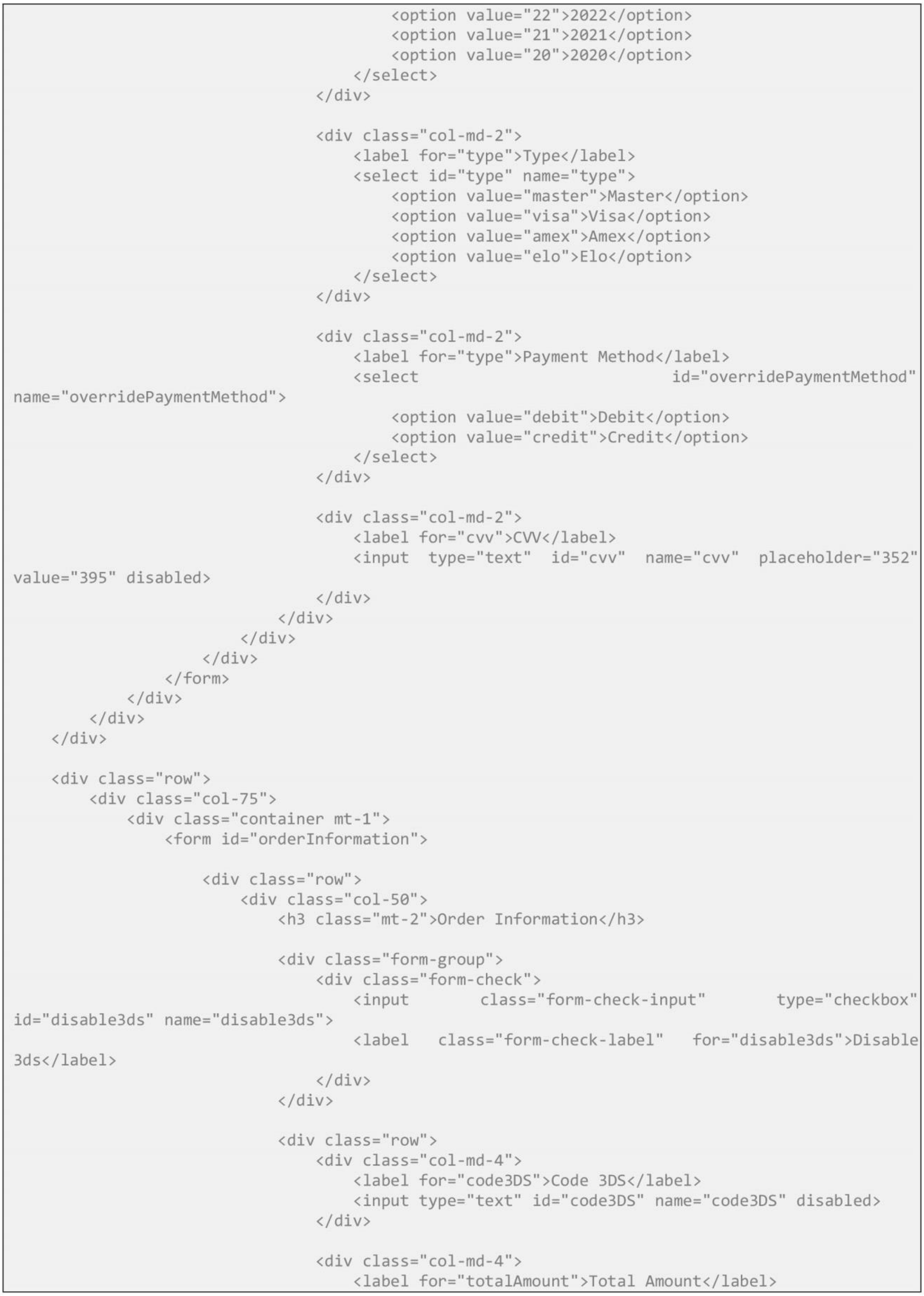


```
index.cshtml

@{
```

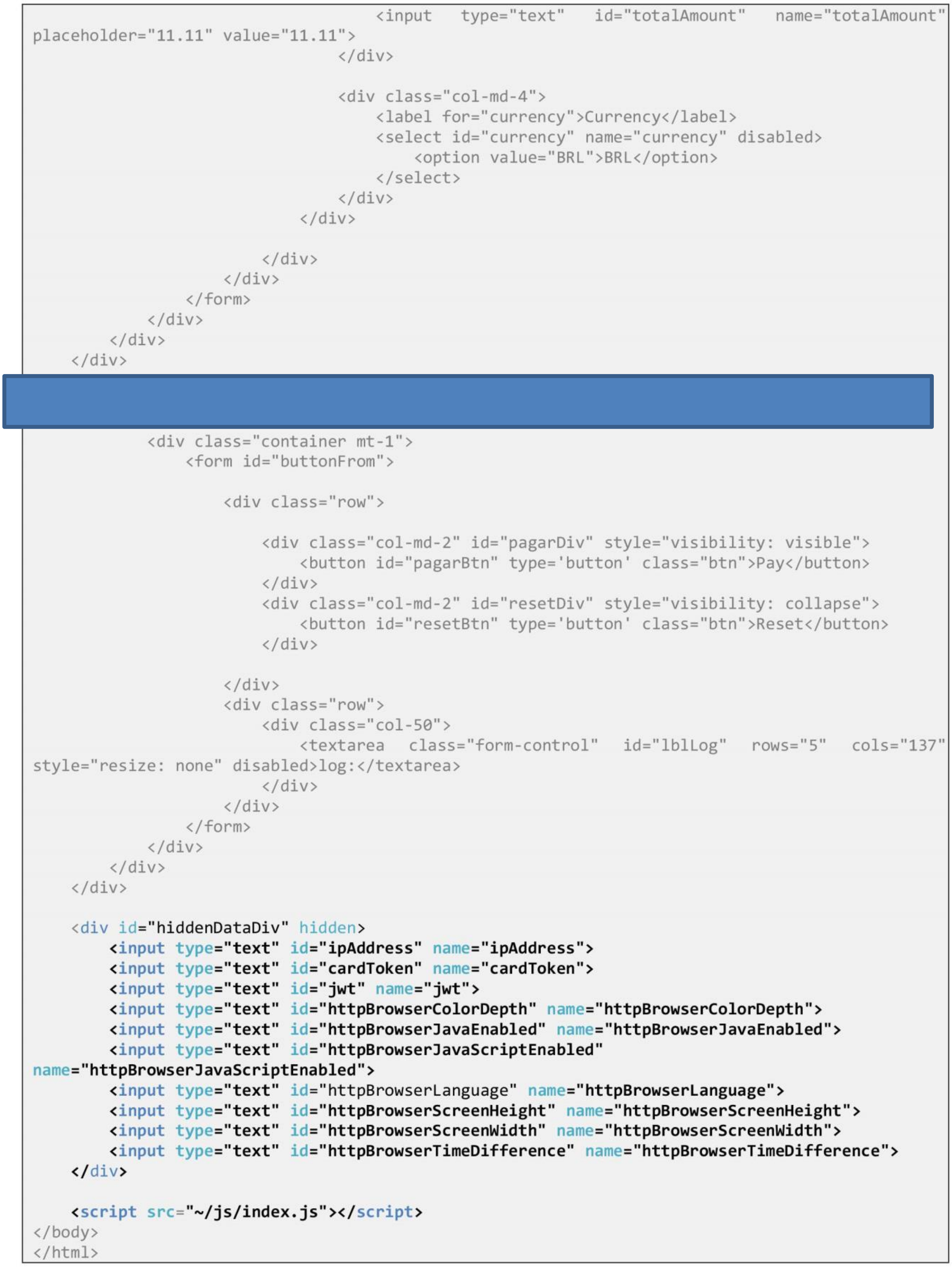
This document is: Public. The information contained in this document must maintain a lower level of protection and may be disclosed without restriction to the general public.



```
    ViewData["Title"] = "Home Page";
}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>3DS</title>
    <script>
        window.payment_url = '@Url.Action("Pagar", "Home")';
    </script>
</head>
<body>
    <div class="row">
        <div class="col-75">
            <div class="container mt-4">
                <form id="cardForm">

                    <div class="form-row">
                        <div class="col-50">
                            <h3 class="mt-2">Payment</h3>
                            <div class="form-row">
                                <div class="col-md-4">
                                    <label for="ccnum">Credit card number</label>
                                    <input     type="text"     id="ccnum"     name="cardnumber"
placeholder="1111-2222-3333-4444" value="5200000000001005">
                                </div>
                                <div class="col-md-8">
                                    <label for="cname">Name on Card</label>
                                    <input     type="text"     id="cname"     name="cardname"
placeholder="John More Doe" value="Luiz Silveira Neto" disabled>
                                </div>
                            </div>

                            <div class="row">
                                <div class="col-md-2">
                                    <label for="expmonth">Exp Month</label>
                                    <select id="expmonth" name="expmonth" disabled>
                                        <option value="01">Janeiro</option>
                                        <option value="02">Fevereiro</option>
                                        <option value="03">Março</option>
                                        <option value="04">Abril</option>
                                        <option value="05">Maio</option>
                                        <option value="06">Junho</option>
                                        <option value="07">Julho</option>
                                        <option value="08">Agosto</option>
                                        <option value="09">Setembro</option>
                                        <option value="10">Outubro</option>
                                        <option value="11">Novembro</option>
                                        <option value="12">Dezembro</option>
                                    </select>
                                </div>

                                <div class="col-md-2">
                                    <label for="expyear">Exp Year</label>
                                    <select id="expyear" name="expyear" disabled>
                                        <option value="29">2029</option>
                                        <option value="28">2028</option>
                                        <option value="27">2027</option>
                                        <option value="26">2026</option>
                                        <option value="25">2025</option>
                                        <option value="24">2024</option>
                                        <option value="23">2023</option>
```



```
                                        <option value="22">2022</option>
                                        <option value="21">2021</option>
                                        <option value="20">2020</option>
                                    </select>
                                </div>

                                <div class="col-md-2">
                                    <label for="type">Type</label>
                                    <select id="type" name="type">
                                        <option value="master">Master</option>
                                        <option value="visa">Visa</option>
                                        <option value="amex">Amex</option>
                                        <option value="elo">Elo</option>
                                    </select>
                                </div>

                                <div class="col-md-2">
                                    <label for="type">Payment Method</label>
                                    <select                           id="overridePaymentMethod"
name="overridePaymentMethod">
                                        <option value="debit">Debit</option>
                                        <option value="credit">Credit</option>
                                    </select>
                                </div>

                                <div class="col-md-2">
                                    <label for="cvv">CVV</label>
                                    <input  type="text"  id="cvv"  name="cvv"  placeholder="352"
value="395" disabled>
                                </div>
                            </div>
                        </div>
                    </div>
                </form>
            </div>
        </div>
    </div>

    <div class="row">
        <div class="col-75">
            <div class="container mt-1">
                <form id="orderInformation">

                    <div class="row">
                        <div class="col-50">
                            <h3 class="mt-2">Order Information</h3>

                            <div class="form-group">
                                <div class="form-check">
                                    <input      class="form-check-input"      type="checkbox"
id="disable3ds" name="disable3ds">
                                    <label   class="form-check-label"   for="disable3ds">Disable
3ds</label>
                                </div>
                            </div>

                            <div class="row">
                                <div class="col-md-4">
                                    <label for="code3DS">Code 3DS</label>
                                    <input type="text" id="code3DS" name="code3DS" disabled>
                                </div>

                                <div class="col-md-4">
                                    <label for="totalAmount">Total Amount</label>
```



```
                                    <input type="text" id="totalAmount" name="totalAmount"
placeholder="11.11" value="11.11">
                                </div>

                                <div class="col-md-4">
                                    <label for="currency">Currency</label>
                                    <select id="currency" name="currency" disabled>
                                        <option value="BRL">BRL</option>
                                    </select>
                                </div>
                            </div>

                        </div>
                    </div>
                </form>
            </div>
        </div>
    </div>
```

[redacted]

```
            <div class="container mt-1">
                <form id="buttonFrom">

                    <div class="row">

                        <div class="col-md-2" id="pagarDiv" style="visibility: visible">
                            <button id="pagarBtn" type='button' class="btn">Pay</button>
                        </div>
                        <div class="col-md-2" id="resetDiv" style="visibility: collapse">
                            <button id="resetBtn" type='button' class="btn">Reset</button>
                        </div>

                    </div>
                    <div class="row">
                        <div class="col-50">
                            <textarea class="form-control" id="lblLog" rows="5" cols="137"
style="resize: none" disabled>log:</textarea>
                        </div>
                    </div>
                </form>
            </div>
        </div>
    </div>

    <div id="hiddenDataDiv" hidden>
        <input type="text" id="ipAddress" name="ipAddress">
        <input type="text" id="cardToken" name="cardToken">
        <input type="text" id="jwt" name="jwt">
        <input type="text" id="httpBrowserColorDepth" name="httpBrowserColorDepth">
        <input type="text" id="httpBrowserJavaEnabled" name="httpBrowserJavaEnabled">
        <input type="text" id="httpBrowserJavaScriptEnabled"
name="httpBrowserJavaScriptEnabled">
        <input type="text" id="httpBrowserLanguage" name="httpBrowserLanguage">
        <input type="text" id="httpBrowserScreenHeight" name="httpBrowserScreenHeight">
        <input type="text" id="httpBrowserScreenWidth" name="httpBrowserScreenWidth">
        <input type="text" id="httpBrowserTimeDifference" name="httpBrowserTimeDifference">
    </div>

    <script src="~/js/index.js"></script>
</body>
</html>
```

5. Required fields and validations in the e-commerce gateway

There is a list of fields that are required to be sent to the Gateway when we are running 3DS. These fields, according to the requisitions, are:

v1/payments (POST)

Field	Description	Type	Size	Mandatory
DeviceInfo. httpAcceptBrowserValue	3DS - Value of the Accept Header of the client's browser.	string	255	Yes
DeviceInfo. httpAcceptContent	3DS - Exact value of HTTP Accept Header.	string	256	Yes
DeviceInfo. httpBrowserLanguage	3DS - Client's browser language according to https://www. techonthenet. com/js/language_tags. php .	string	8	Yes
DeviceInfo. httpBrowserJavaEnabled	3DS - If JAVA enabled send Y value, otherwise N.	string	1	Yes
DeviceInfo.httpBrowserJavaScriptEnabled	3DS - If JAVA SCRIPT enabled send Y value, otherwise N.	string	1	Yes
DeviceInfo. httpBrowserColorDepth	3DS - Number of bits used to display images.	string	3	Yes
DeviceInfo. httpBrowserScreenHeight	3DS - Height of the client's screen resolution.	string	4	Yes
DeviceInfo. httpBrowserScreenWidth	3DS - Width of the client screen resolution.	string	4	Yes
DeviceInfo. httpBrowserTimeDifference	3DS - Difference in minutes between GMT time and that of the client's browser.	string	4	Yes
DeviceInfo. userAgentBrowserValue	3DS - The exact value of the User Agent Header.	string	100	Yes

3DS data

All the fields above are in the Body.

Field	Description	Type	Size	Mandatory
SellerInfo.code3DS	Merchant 3DS code. If not sent, 3DS will not be executed.	guid	36	Yes
SellerInfo. urlSite3DS	URL of the commerce website used for 3ds, for example, the website is https://minhaempresa.com.br so you must inform [minhaempresa. com. br] without informing [**Erro! A referência de hiperlink não é válida.**].	string	2048	Yes

Customer's data

All the fields above are in the Body.

Field	Description	Type	Size	Mandatory
Customer. DocumentType	Customer's identification document type (CPF, CNPJ).	int	1	No
Customer. DocumentNumber	Customer's document number without punctuation (without mask).	string	20	No

Customer. FirstName	First name of the customer.	string	60	Yes
Customer. LastName	Last name of the customer.	string	60	Yes
Customer. Email	Customer's email.	string	255	Yes
Customer. PhoneNumber	Customer's phone (without mask).	string	15	Yes
Customer. mobilePhoneNumber	Customer's cell phone (without mask).	string	25	Yes
Customer. Address	Customer’s address.	string	60	Yes
Customer. Complement	Complement of the customer's address.	string	60	No
Customer. City	Customer’s city.	string	50	Yes
Customer. State	Customer's state.	string	20	Yes
Customer. ZipCode	Customer's zip code (without mask).	string	10	Yes
Customer. IpAddress	IP address of the customer's device.	string	48	Yes
Customer.country	Country of the customer's address.	string	2	Yes

All the fields above are in the Body.

Json v1/payments Request

```
{
    “Payment”:{
      “TransactionType”: “debit”,
      “Amount”
      “CurrencyCode”: “brl”,
      “ProductType”: “debito”,
      “Installments”: 1,
      “CaptureType”: “ac”,
      “Recurrent”:false
    },
    “Cardlnfo”: {
      “NumberToken”:”d56109ce-6a3b-4190-beea-1c975e8cd48 6”,
      “CardholderName”:”Luiz Silveira Neto”,
      “SecurityCode”:”123”,
      “Brand”: “mastercard”,
      “ExpirationMonth”:”01”,
      “ExpirationYear”:”29”
    },
    “SellerInfo”:{
      “OrderNumber”:”1110197548565”,
      “SoftDescriptor”:”PAG*TESTE”,
      “UrlSite3DS”:”empresa.ecommerce.com.br”,
      “Code3DS”:”aa415dc21f4e4e7ca0 8 3a147 3d5c4c5 8”
    },
    “Customer”:{
      “DocumentType”: “cpf”,
      “DocumentNumber”:”51115672088”,
      “FirstName”:”Luiz”,
      “LastName”:”Silveira Neto”,
      “Email”:”luiz.silveira@teste.rafael.com“,
      “PhoneNumber”:”1122542454”,
      “MobilePhoneNumber”:”11987683332”,
      “Address”:”Rua Luiz Vieira, 134”,
      “Complement”:”apto. 34 - Vila Guarani”,
      “City”:”São Paulo”,
      “State”:”SP”,
      “ZipCode”:”0987 6-098”,
      “IpAddress”:”45.233.2 32.248”,
      “Country”:”BR”
    },
    “Sellers”:[

    ] ,
    “Devicelnfo”:{
“HttpAcceptBrowserValue”:”text/html,application/xhtml+xml,application/xml;
q=0.9,image/webp,image/apng,*/*;q=0.8”,
“HttpAcceptContent”:”text/html,application/xhtml+xml,application/xml;q=0.9
, image/webp,image/apng,*/*; q=0.8”,
      “HttpBrowserLanguage”:”pt-BR”,
      “HttpBrowserJavaEnabled”:”N”,
      “HttpBrowserJavaScriptEnabled”:”Y”,
      “HttpBrowserColorDepth”:”24”,
      “HttpBrowserScreenHeight”:”937”,
      “HttpBrowserScreenWidth”:”1920”,
      “HttpBrowserTimeDifference”:”180”,
    “UserAgentBrowserValue”:”Mozilla/5.0 (Windows NT 10.0; Win64; x64)
 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.102 Safari/537.36”
    }
}
```

If a Challenge occurs, "v1/payments" will return the following message through "ThreeDs". Check the JSON below:

Response:

```
{
   “threeDs”:{
       “status”:”Challenge”,
       “threeDsVersion”:”2.1.0”,
“acsUrl”:”https://0merchantacsstag.cardinalcommerce.com/MerchantACSWeb/
cre q.jsp “,
   “authenticationTransactionId”:”BGprC3ktx8lMtL391Ma0”,
“pareq”:”eyJtZXNzYWdlVHlwZSI6IkNSZXEiLCJtZXNzYWdlVmVyc2lvbiI6IjIuMS4wIi
widGhyZWVEU1NlcnZlclRyYW5zSUQiOiI4N2YwYzdjOS0wODlkLTQ5OGItYTYzZC03MjJiM
zc5ZTM wOTEiLCJhY3NUcmFuc0lEIj oiNGEzOWMxNGEtZjBiNi 0
0YTE5LTg3MWEtYmQ2MjZlZDcyNmEzI iwiY2hhbGxlbmdlV2luZG93U2l6ZSI6IjAyIn0”
```

```
        }
    }
```

Ownership	Description	Type	Mandatory
ThreeDs.Status	3DS Challenge Status - 3DS is requesting a Challenge	string (30)	Yes
ThreeDs.ThreeDsVersion	ThreeDsVersion is the 3DS version and will be informed when it is Challenge	string(l0)	Yes
ThreeDs. AcsUrl	AcsUrl is the return when it is Challenge, you must inform it inAdiq3ds.InitChallenge (acsUrl, pareq, authenticationTransactionId)	string(200)	Yes
ThreeDs. Pareq	Pareq is the return when it is Challenge, you must inform it inAdiq3ds.InitChallenge (acsUrl, pareq, authenticationTransactionId)	string(800)	Yes
ThreeDs. AuthenticationTransactionId	AuthenticationTransactionId is the return when it is Challenge, you must inform itinAdiq3ds.InitChallenge (acsUrl, pareq, authenticationTransactionId)	string(40)	Yes

All the properties above are in the Body.

Payment validation with 3DS Challenge

To validate a transaction with 3DS challenge, it is necessary to send a request using the POST method for the "Validate" resource, according to the example:

Request:

/v1/payments/validate (POST)

Ownership	Description	Type	Mandatory	Size	Mandatory
Code3DS	3DS identification code.	guid	yes	36	Yes

ValidateToken	Validation token generated by 3DS when returning a challenge.	string	yes	500	Yes

Json v1/payments/validate Request

```
{
        “code3ds”: “dbe5a61a-836d-45dd-82c1-52d0c5e3c010”,
        “validateToken”:
“eyJhbGciOiJIUzI1NiJ9.eyJpc3MiOiI1ZWNkNzVkNjNlMzFiYzQ3MWExZWM2MWYiLCJpYXQi
OjE1OTQyMzI4MTcsImV4cCI6MTU5NDI0MDAxNywianRpIjoiYj FmNjEyZWEtMDRhZS0
0ZDY2LW EzM2QtY2I1MjgwOGY4M2FlIiwiQ2 9uc3VtZXJTZXNzaW9uSWQiOiIwX2E3MGIzMj
k0LTI1MTItNDc0ZS05OTcxLTZmZTk3YTllYWYzNiIsIlJlZmVyZW5jZUlkIjoiN2Y0ODc3YmVk
ODVhNDU2MmIxYjNlZjk4MWQ2MjViY2UiLCJhdWQiOiIwMzc2NTk3ODgyMTU0NWRiOTBjN2Y1Ym
E1ZDNmNWU1NiIsIlBheWxvYWQiOiJ7XCJQYXltZW50XCI6e1wiVHlwZVwiOlwiQ0NBXCIsXCJQ
cm9jZXNzb3JUcmFuc2FjdGlvbklkXCI6XCJHc2pMM1lkczVmWk80MzBXYk1RMFwiLFwiRXh0ZW
5kZWREYXRh XCI6e1wiQW1vdW5 0XCI6XCIxMTExXCIsXCJDdXJyZW5j eUNvZGVcIjpcIj
k4NlwifX0sXCJFcn Jvck51bWJlclwiOjAsXCJFcnJvckRlc2NyaXB0aW9uXCI6XCJTdWNj
ZXNzXCJ9In0.9oBbxbtu hZyZaE_vuu6UzU-Mp5Bj K2 FNp5XAfoDy2uQ” }
```

Json Response Code:200 OK

```
{
   “paymentAuthorization”:{
       “returnCode”:”0”,
       “description”:”Sucesso”,
       “paymentId”:”02 000 62 0440 92 31653310000 02 3790740000000000”,
       “authorizationCode”:”026603”,
       “orderNumber”:”3979908797721”,
       “amount”:1111,
       “releaseAt”:”2 02 0-0 9-2 3T16:53:34+0 0:00”
   }
}
```

Json Response Code: 400 BadRequest

```
{
   “FailureDetails”:[
      {
          “tag”: “string”,
          “description”: “motivo da falha”
      }
   ]
}
```

All the properties above are in the Body.

Domains

Ownership	Contents
Payment.TransactionType	credit, debit
Payment.CurrencyCode	brl
Payment.ProductType	avista, lojista, debito
Payment.CaptureType	ac - Authorizes and captures, pa - Pre-authorizes
Payment.Recurrent	true - Recurring, false - Not recurring
Cardinfo.Brand	visa, mastercard, amex, elo, hipercard

CyberSource Response Domains

- **consumerAuthenticationInformation.veresEnrolled**
  - Y – Yes, Bank is participating in 3-D Secure protocol and will return the ASCUrl
  - N – No, Bank is not participating in 3-D Secure protocol
  - U – Unavailable, The DS or ACS is not available for authentication at the time of the request.
  - B – Bypass, Merchant authentication rule is triggered to bypass authentication in this use case

- **consumerAuthenticationInformation.paresStatus**
  - Y – Successful Authentication
  - N – Failed Authentication / Account Not Verified / Transaction Denied
  - U – Unable to Complete Authentication
  - A – Successful Attempts Transaction
  - C – Challenge Required for Authentication
  - R – Authentication Rejected (Merchant must not submit for authorization)
  - D – Challenge Required, Decoupled Authentication confirmed
  - I – Information Only, 3DS Requestor challenge preference acknowledged

- **consumerAuthenticationInformation.eciRaw**
  - 02 or 05 – Fully Authenticated Transaction
  - 01 or 06 – Attempted Authentication Transaction

  - 00 or 07 – Non 3-D Secure Transaction

- **consumerAuthenticationInformation.cardholderMessage**
  - Message provided by the ACS during a frictionless authentication process.

- **status**
  - AUTHENTICATION_SUCCESSFUL
  - PENDING_AUTHENTICATION
  - INVALID_REQUEST
  - AUTHENTICATION_FAILED

- **errorInformation.reason**
  - INVALID_MERCHANT_CONFIGURATION
  - CONSUMER_AUTHENTICATION_REQUIRED
  - CONSUMER_AUTHENTICATION_FAILED
  - AUTHENTICATION_FAILED

6. Capture browser fields via JavaScript

Below is a suggestion to capture the fields below via JavaScript:

1. **httpAcceptBrowserValue;**
   a. Which extensions the browser can accept;

Example:

*text/html, application/xhtml+xml, application/xml;q=0. 9, image/webp, image/apng, */*;q=0. 8*

2. **httpAcceptContent;**
   a. What content will be accepted;

Example:

*text/html, application/xhtml+xml, application/xml;q=0. 9, image/webp, image/apng, */*;q=0.*

8

3. **`httpBrowserLanguage`**
   a. `var httpBrowserLanguage = navigator.language || navigator.userLanguage;`

4. **`httpBrowserJavaEnabled`**
   a. `navigator.javaEnabled() == true ? “Y” : “N”;`

5. **`httpBrowserJavaScriptEnabled`**
   a. `navigator.javaEnabled() == true ? “Y” : “N”;`

6. **httpBrowserColorDepth**
    a. screen. colorDepth

7. **httpBrowserScreenHeight**
    a. window. innerHeight

8. **httpBrowserScreenWidth**
    a. window. innerWidth

9. **httpBrowserTimeDifference**
    a. new Date().getTimezoneOffset()

10. **userAgentBrowserValue**
    a. navigator. userAgent

7. File dictionary

Files that are used in this process

index.js: Payment page file (Front-End).

Layout.cshtml: Where the JQuery script is placed. If the user of this documentation uses React, it will be necessary to place in the file where the JQuery script is placed.

Note: DTO files exchange information from Front to Backend.

8. Dictionary of names

Names that are used in this process

ccnum: Card number.

validateChallengeCallback: Method that receives response from the Challenge. The parameter of this method are two strings

- being the first parameter ValidateToken that must be sent to v1/payments/validate
- the second parameter is the status of the challenge [Approved/Cancelled],
    - If Canceled, the process must be interrupted because there was a cancellation of the challenge process.

treatEnrollmentResponse: Method that will receive the payment response (in index.js - front-end).

CallPayment: Payment call.