



OPEN SOURCE

TECH CONFERENCE



M.



oi :)



Guilherme Coelho



Pagar.me



Jundiaí





Pagar.me Checkout



Overview do Produto



O que é o Pagar.me Checkout?



Overview do Produto

1. Checkout transparente
2. Captura os dados do pagador
3. Experiência do Usuário
4. Checkout customizável
5. Autoriza transação no banco



Informe seus dados de cartão

Total a pagar R\$143,50



": "credit_card",
null,
123083_ruok/koB4wbvc/fwZ81eLIVRyy2/111T6m1x0t017mzvnn1wqos9xt/1Ppa6EURLmKoC/LEkhVLzwL2

3479 250472 58652



Nicolas Cage



03/19 ✓

1234 ✓

3x



Pagar (R\$143,50)



Informe seus dados de cartão

Total a pagar R\$143,50



```
": "credit_card",
  null,
i23083_ruoK/koB4wbvc/fWZ91eUVEvzJ/0HUTAnIuL017pnwpmD4C6S2CVTlPoa6EURLmKoC/LEkHvLzwL2
```

3479 250472 58652



Nicolas Cage



03/19



1234



3X



Pagar (R\$143.50)



bit.ly/ostc2a



Como utilizar?



Via formulário, via checkout api



Utilizando via formulário

```
<form method="POST" action="/comprar">
  <script type="text/javascript"
    src="https://assets.pagar.me/checkout/checkout.js"
    data-button-text="Pagar um café"
    data-encryption-key="ek_test_*****xqRmk"
    data-amount="2025">
  </script>
</form>
```



Utilizando via API

```
const button = document.querySelector('#opensource-day')
button.addEventListener('click', openCheckout)

function openCheckout () {
  const checkout = new PagarMeCheckout.Checkout({
    encryption_key: "ek_test_*****xqRmk",
    success: () => alert('Transação autorizada! :)')
  })

  checkout.open({
    amount: 2025,
    installments: 2,
    payment_method: 'credit_card'
  })
}
```



Use Cases



Modo token, modo card_hash





Modo Token

data-create-token="true"



Modo Token

1. Cria transação de cartão e autoriza no banco
2. Retorna um token da transação
3. Não tem captura automática
4. Autorização dura **5 horas**



Modo Token

```
{  
  amount: 14350,  
  payment_method: 'credit_card',  
  createToken: true // default  
}
```

Retorno:

```
{  
  "token": "test_transaction_CS2Eixp1aVYNfzCnVtLzkd1iUiZ8b3",  
  "payment_method": "credit_card"  
}
```



Modo CardHash

`data-create-token="false"`



Modo CardHash

1. Não cria transação
2. Retorna o card_hash (que representa os dados de cartão)
3. card_hash pode ser usado **apenas 1 vez**
4. card_hash tem duração de **5 minutos**
5. Use case: assinaturas



Modo CardHash

```
{  
  amount: 14350,  
  payment_method: 'credit_card',  
  createToken: false  
}
```

Retorno:

```
{  
  "payment_method": "credit_card",  
  "amount": 14350,  
  "installments": "4",  
  "card_hash": "123456_1aNbu*****VqnNWpMA"  
}
```



Detalhes Técnicos



Como o checkout foi construído?



Endpoints da API

Quais endpoints da API usamos no checkout?



Chamada de endpoint da API

```
GET https://api.pagar.me/1/endpoint
```

```
{  
  encryption_key: ek_test_*****2xa4b  
}
```



Preencher endereço do cliente

```
GET /zipcodes/04551902
```

```
{  
  "zipcode": "04551902",  
  "street": "Rua Fidêncio Ramos 308",  
  "neighborhood": "Vila Olímpia",  
  "city": "São Paulo",  
  "state": "SP"  
}
```



Calcular pagamentos parcelados

GET /transactions/calculate_installments

```
{  
  "installments": {  
    "1": {  
      "installment": 1,  
      "amount": 10000,  
      "installment_amount": 10000  
    },  
    "2": {  
      "installment": 2,  
      "amount": 10598,  
      "installment_amount": 5299  
    }  
  }  
}
```



Criar card_hash

GET /transactions/card_hash_key

```
{  
  "date_created": "2015-02-27T15:44:26.000Z",  
  "id": 111111,  
  "ip": "000.0.00.00",  
  "public_key": "-----BEGIN PUBLIC KEY-----\n-----END PUBLIC KEY-----"\n}
```



Criar transação

POST /transactions

```
{  
  "status": "authorized",  
  "amount": 1000,  
  "payment_method": "credit_card",  
  "capture_method": "ecommerce",  
  "referer": "encryption_key",  
  ...  
}
```



pagarme-js (lib)

[pagarme / pagarme-js](#)

[Code](#) [Issues 6](#) [Pull requests 3](#) [Projects 0](#) [Wiki](#) [Settings](#) [Insights](#)

Pagar.me's Javascript API [Edit](#)

Add topics

209 commits 20 branches 19 releases 7 contributors MIT

Branch: master ▾ New pull request Create new file Upload files Find file Clone or download ▾

MarcoWorms release: version 3.1.0 Latest commit 098760c on May 8

browser	release: version 3.1.0	2 months ago
dist	tests: move session to new folder structure	4 months ago
lib	tests: fix e2e tests	2 months ago
test	request: add qs to put request because some of our endpoints need it ...	3 months ago
.babelrc	chore: add ramda babel plugin	3 months ago
.eslintrc	Add comma dangle for diff happiness	6 months ago
.gitignore	chore: fix doc generation	3 months ago

<https://github.com/pagarme/pagarme-js>



Tecnologia



O que usamos para construir o checkout



Tecnologia

1. Modo "clássico" de desenvolvimento web
2. Javascript + jQuery
3. CSS + Sass
4. Gulp
5. Tests: Docker + phantomJS



Arquitetura



Como o checkout funciona internamente



Arquitetura

1. Arquivos: checkout.js e modal.js
2. Steps de pagamento (customer_data, credit_card)
3. Execução dentro de iframe (via CDN)
4. Não tem versionamento (pros e cons)



Comunicação Iframe

Comunicação host - iframe - pagarme

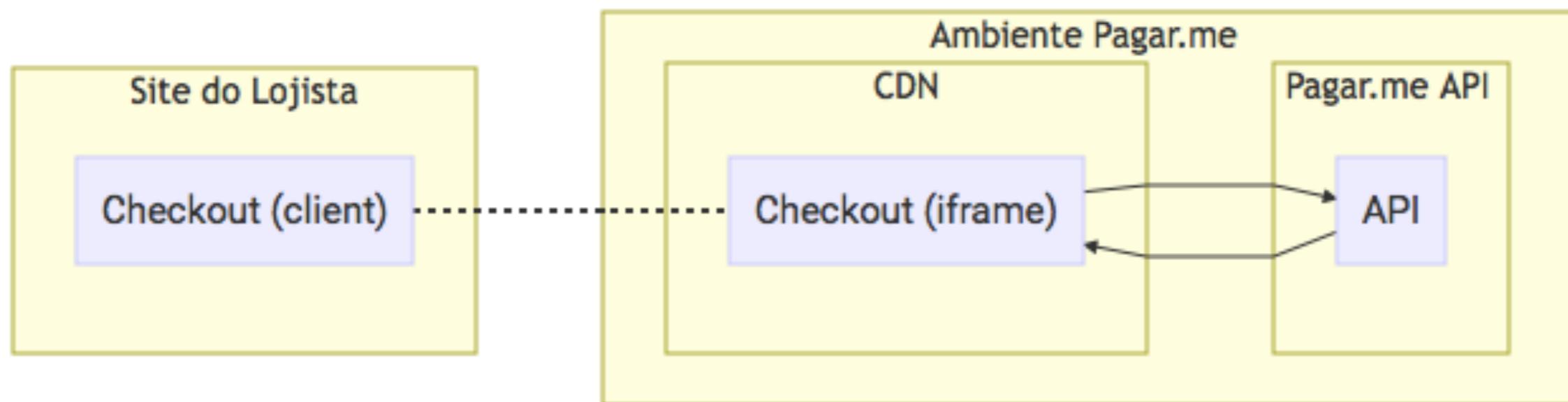


Comunicação Iframe

1. Comunicação via tag <title>
2. Suporte IE6+
3. easyXDM: lib de comunicação entre iframe e host
4. Client não precisa de HTTPS (embora seja recomendado)



Iframe - fluxo de comunicação



Roadmap

Onde estamos, para onde vamos



Onde estamos

Jun 8, 2014 – Jun 21, 2017

Contributions to master, excluding merge commits

Contributions: Commits ▾



Onde estamos

1. Funciona bem para as features que já estão implementadas
2. Código legado + Tecnologia Antiga
3. Dificuldade de debuggar problemas
4. Dificuldade de implementar novas features



Para onde vamos? 

O futuro do Checkout



Reconstruir o Checkout

Tecnologia, design e experiência modernas





OPEN SOURCE



Vamos fazer juntos? 💪



Links

Github, Documentação, etc



Github

[pagarme / pagarme-checkout](#) Private

[Unwatch](#) 23 [Star](#) 0 [Fork](#) 2

[Code](#) [Issues 2](#) [Pull requests 1](#) [Projects 0](#) [Wiki](#) [Settings](#) [Insights](#)

Pagar.me checkout form [Edit](#)

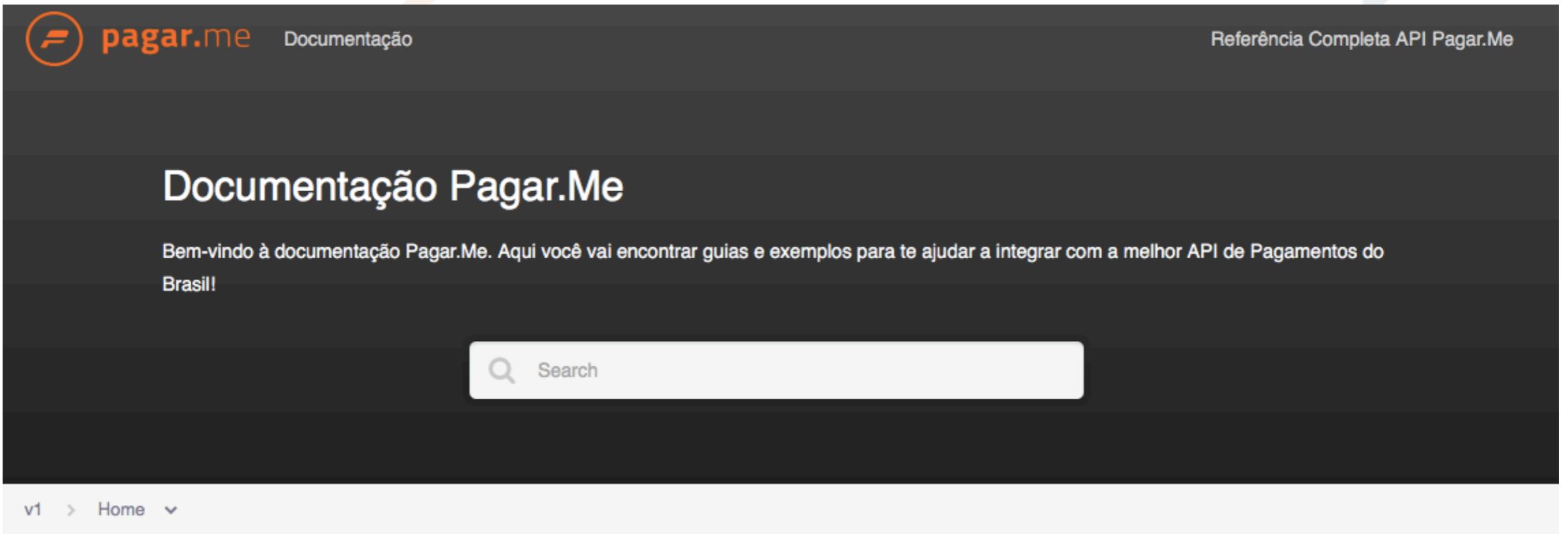
[Add topics](#)

[565 commits](#) [17 branches](#) [41 releases](#) [14 contributors](#)

Branch: [master](#) [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#) [Clone or download](#)

<https://github.com/pagarme/pagarme-checkout>

Documentação



The screenshot shows the homepage of the Pagar.Me documentation. At the top left is the Pagar.Me logo with the text "pagar.me" next to it. To its right is the word "Documentação". On the far right of the header is a link "Referência Completa API Pagar.Me". The main title "Documentação Pagar.Me" is centered above a brief welcome message: "Bem-vindo à documentação Pagar.Me. Aqui você vai encontrar guias e exemplos para te ajudar a integrar com a melhor API de Pagamentos do Brasil!". Below the message is a search bar with the placeholder "Search". At the bottom left of the page, there is a navigation bar with "v1" and "Home" followed by a dropdown arrow.

<https://docs-beta.pagar.me/docs/overview-checkout>

Obrigado ❤



**Guilherme Rv
Coelho**
grvcoelho

While I'm not 🌟 billionaire 🌟 I do things such as coding, juggling and magic for food. Working with ❤ for @pagarme

👤 Pagar.me
📍 Jundiaí, Brazil
🔗 <https://pagarme>

Overview Repositories 24 Stars 56 Followers 146 Following 36

Pinned repositories

≡ react-voice-components

Set of React components that use the Web Speech API to bring voice experience to React applications

JavaScript ★ 124 ⚡ 12

≡ starhopper

🔥 Front-end boilerplate with Webpack, Hot Reload, Tree-Shaking and React

JavaScript ★ 66 ⚡ 8

≡ babelfish

-translate Straightforward library for translations and dictionaries

JavaScript ★ 44 ⚡ 1

Customize your pinned repositories

≡ vault

🔐 Manage credentials securely using AWS KMS and AWS DynamoDB

HCL ★ 1

≡ css-styleguide

📘 Opinionated CSS styleguide for scalable applications

★ 369 ⚡ 25

≡ vim-javascript-snippets

🔧 JavaScript & NodeJS Snippets for Vim

★ 42 ⚡ 3

<https://github.com/grvcoelho>

