Development of a mobile payment service with focus on evaluating API recording for automated mocking
The problem

1 Yuán = 10 Jiǎo = 100 Fēn
1 Dinar = 1000 Fils
1 Ouguiya = 5 Khoums
Structure

- **Overview**
  - The company
  - How do people pay in games?
  - Mobile payment
  - Software as a service (SaaS)
  - Tier based approach

- **Implementation**
  - Two types of providers
  - Tiny little differences

- **How to test?**
  - Mock request vs. recorded request
  - Simulate a real browser
  - Solution
Softgames GmbH

- Founded 2007
- Started with creating and distributing java games
- Expanding to android and html5
- They are publishing HTML5 games
How do people pay in games?
Mobile payment

PayPal

VISA

Google wallet

MasterCard
People are playing on their mobile
Use the mobile to pay
Mobile payment

### France: Payouts

<table>
<thead>
<tr>
<th>Price (EUR) incl. VAT</th>
<th>0.35</th>
<th>0.50</th>
<th>1.00</th>
<th>1.50</th>
<th>2.00</th>
<th>3.00</th>
<th>3.99</th>
<th>4.50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price (EUR) excl. VAT</td>
<td>0.29</td>
<td>0.42</td>
<td>0.84</td>
<td>1.25</td>
<td>1.67</td>
<td>2.51</td>
<td>3.34</td>
<td>3.76</td>
</tr>
</tbody>
</table>

### Germany: Payouts

<table>
<thead>
<tr>
<th>Price (EUR) incl. VAT</th>
<th>0.49</th>
<th>0.99</th>
<th>1.99</th>
<th>2.99</th>
<th>3.99</th>
<th>4.99</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price (EUR) excl. VAT</td>
<td>0.41</td>
<td>0.83</td>
<td>1.67</td>
<td>2.51</td>
<td>3.35</td>
<td>4.19</td>
</tr>
</tbody>
</table>

### Spain: Payouts

<table>
<thead>
<tr>
<th>Price (EUR) incl. VAT</th>
<th>1.45</th>
<th>3.63</th>
<th>5.20</th>
<th>7.26</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price (EUR) excl. VAT</td>
<td>1.20</td>
<td>3.00</td>
<td>4.30</td>
<td>6.00</td>
</tr>
</tbody>
</table>
Extract the payment from the current product and build it as a service
How can we simplify worldwide payment for developers?
### App Store Pricing Matrix

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Tier 1</td>
<td>0.99</td>
<td>0.70</td>
<td>0.99</td>
<td>0.70</td>
<td>0.79</td>
<td>0.48</td>
<td>0.59</td>
<td>0.36</td>
<td>115</td>
<td>81</td>
<td>1.19</td>
<td>0.76</td>
</tr>
<tr>
<td>Tier 2</td>
<td>1.99</td>
<td>1.40</td>
<td>1.99</td>
<td>1.40</td>
<td>1.59</td>
<td>0.97</td>
<td>1.19</td>
<td>0.72</td>
<td>230</td>
<td>161</td>
<td>2.49</td>
<td>1.58</td>
</tr>
<tr>
<td>Tier 3</td>
<td>2.99</td>
<td>2.10</td>
<td>2.99</td>
<td>2.10</td>
<td>2.39</td>
<td>1.45</td>
<td>1.79</td>
<td>1.09</td>
<td>350</td>
<td>245</td>
<td>3.99</td>
<td>2.54</td>
</tr>
<tr>
<td>Tier 4</td>
<td>3.99</td>
<td>2.80</td>
<td>3.99</td>
<td>2.80</td>
<td>2.99</td>
<td>1.82</td>
<td>2.39</td>
<td>1.45</td>
<td>450</td>
<td>315</td>
<td>4.99</td>
<td>3.18</td>
</tr>
<tr>
<td>Tier 5</td>
<td>4.99</td>
<td>3.50</td>
<td>4.99</td>
<td>3.50</td>
<td>3.99</td>
<td>2.43</td>
<td>2.99</td>
<td>1.82</td>
<td>600</td>
<td>420</td>
<td>5.99</td>
<td>3.81</td>
</tr>
<tr>
<td>Tier 6</td>
<td>5.99</td>
<td>4.20</td>
<td>5.99</td>
<td>4.20</td>
<td>4.99</td>
<td>3.04</td>
<td>3.49</td>
<td>2.12</td>
<td>700</td>
<td>490</td>
<td>7.99</td>
<td>5.08</td>
</tr>
<tr>
<td>Tier 9</td>
<td>8.99</td>
<td>6.30</td>
<td>8.99</td>
<td>6.30</td>
<td>7.99</td>
<td>4.25</td>
<td>5.49</td>
<td>3.34</td>
<td>1000</td>
<td>700</td>
<td>11.9</td>
<td>7.63</td>
</tr>
<tr>
<td>Tier 10</td>
<td>9.99</td>
<td>7.00</td>
<td>9.99</td>
<td>7.00</td>
<td>8.99</td>
<td>4.86</td>
<td>5.99</td>
<td>3.65</td>
<td>1200</td>
<td>840</td>
<td>12.99</td>
<td>8.27</td>
</tr>
</tbody>
</table>

Implementation

- Ruby on Rails
- Postgres
- GitHub
- CoffeeScript
- Heroku
How can we simplify provider implementation?
two different types of providers
Implementation

● JavaScript
  ○ Insert a script into the DOM of the page
  ○ Insert a signed object into the page
  ○ Execute a function after the page was loaded

● Redirect
  ○ Create query parameters
  ○ Sign those parameters
  ○ Redirect the user to providers page
How to test this?
Mock all request?
Simulate a real browser?
Record and playback requests
Test with a simulated browser against a record version of the provider page?
Testing

- Development of a proxy server
  - Problems with HTTPS
  - Not really usable in our case
- Solution
  - Mix of recorded and mocked request
Thanks!