

Overview:

One of the most important things Paytient does is give our users control over how they pay for their out-of-pocket medical expenses. Usually this happens through payroll deductions or automatic bank drafts, but imagine you are a new engineer on our Payments team and that after months of feature request upvotes, we are finally going to implement one-time payments.

Your objective is to write a new API endpoint, `POST /one-time-payment`, that accepts any payment amount greater than \$0 and returns a user's updated balance and their next payment due date, which is always 15 days into the future. We are also experimenting with a new matching program where Paytient will apply a "match" payment as a percentage of the one-time payment to the remaining balance. Use the table below to determine what match to apply for a tiered system of one-time payments.

One-time Payment Amount	Match
$0 \leq x < 10$	1%
$10 \leq x < 50$	3%
$x \geq 50$	5%

Lastly, no one can have a payment due on a weekend (since that's a great way to ruin a weekend), so any payment due dates that fall on a Saturday or Sunday should be moved out to the next Monday.

For example, imagine someone with a \$100 balance that makes a \$10 one-time payment on March 14, 2022. They will have a new balance of \$89.70 (their \$10 payment plus 3% of \$10, which is \$0.30) and a next payment due date of March 29, 2022.

Similarly, someone with a \$500 balance that makes a \$75 one-time payment on April 8, 2022 will have a new balance of \$421.25 (their \$75 payment plus 5% of \$75, which is \$3.75) and a next payment due date of April 25, 2022.

What we're looking for:

- Software development fundamentals (we want to see an understanding of test automation, code reuse, and the fundamentals of whichever programming language you choose)
- System design (we're looking for logical method structure, domain modeling, and use of design patterns)
- Communication (your code should be easy to understand and reason about)

What you can expect:

- About 2 hours of your time. We focus on outcomes, not output, but want to set your expectations.
- 48-hour response time (like this challenge, if you submit on a weekend the timer starts the following Monday)